Modicon Hand-held Programmer User's Manual

GM-MICR-HHP

043502912 Version 1.00
August, 1993

MODICON, Inc., Industrial Automation Systems
One High Street
North Andover, Massachusetts 01845

Preface

The data and illustrations in this book are not binding. We reserve the right to modify our products in line with our policy of continuous product improvement. Information in this document is subject to change without notice and should not be construed as a commitment by Modicon, Industrial Automation Systems. Modicon, Inc. assumes no responsibility for any errors that may appear in this document.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, without the express written permission of Modicon, Inc., Industrial Automation Systems. All rights reserved.

Modbus is a trademark of Modicon, Inc.

MODSOFT® and MODSOFT® Lite are registered trademarks of Modicon, Inc.

Copyright© 1993 by Modicon, Inc. All rights reserved.

Contents

Chapter 1 Introduction to the Hand-held Programmer 1
About this Book 2
Organization 2
Product Capabilities 2
A Panel Hardware Overview 3
The Liquid Crystal Display 3
The Multipurpose LED 3
Communication and Power Connections 4
The Security Keyswitch 4
The Keypad 5
Screen Navigation and Keystroke Dynamics 6
Moving through Data Fields in an HHP Screen 6
Selecting Items from a Scrolled List in a Field 6
Entering Numeric Data in a Field 6
Moving between HHP Screens 7
The Look of the Cursor 8
Providing Power to the HHP 9
Connecting to a Micro PLC 9
Using an Independent Power Supply 10
Hardware Specifications 11
Environmental 11
Electrical 11

Chapter 2 Getting Started 13
Powering Up the HHP 14
Communicating with a PLC 14
If the HHP Cannot Establish Communication 15
If the HHP Fails its Start-up Diagnostics 17
Selecting a Cursor Display 17
Configured and Unconfigured Micro PLCs 18
Configuration Modes 18

Modes of Communication with a Configured PLC 19
Top-level PLC Operating Modes 19
Operating the PLC 22
Accessing the Operations Screens 22
Starting a Stopped PLC 22
Stopping a Running PLC 22
Clearing User Logic from PLC Memory 23

Chapter 3 Configuring a Micro PLC 25
The PLC Config Options 26
The Three Config Options 26
Configuration Options for an Unconfigured PLC 26
Configuration Options for a Previously Configured PLC 27
How to Autoconfigure a Micro PLC 28
Accessing Autoconfig when the PLC Is Unconfigured 28
Accessing Autoconfig when the PLC Is Configured 28
Autoconfiguring a PLC in Single Mode 29
Autoconfiguring a PLC in Child Mode 30
Autoconfiguring a PLC in Parent Mode 31
Customizing a PLC Configuration 33
Config Parameters that Can Be Edited 33
Accessing the Set config Screens 33
Editing Comm Parameters on an RS-232 Port 33
Setting Messaging Parameters 36
Viewing Configuration Parameters 39
Viewing an Unconfigured PLC 39
Viewing a Configured PLC 39

Chapter 4 Addressing I/O Locations with the I/O Map Editor 43
Invoking the I/O Map Editor 44
Accessing the I/O Map 44
I/O Map Editor Conventions 45
Default I/O Map Parameters 48
Default Parameters for Single and Parent PLCs 48
Default Parameters for A120 I/O LOCs 49
Default Parameters for a Child PLC 50
Default Parameters for a Child PLC Addressed through its Parent 50
Editing and Saving I/O Addresses 51
Accessing the LOC Screens 51
Editing References In an I/O Map Screen—An Overview . . . 51

Saving Changes in the I/O Map Editor 52
Addressing a PLC's Fixed I/O Locations 54
Editing LOC01 54
Editing LOC02 57
Editing LOC03 58
Editing LOC04 60
Sharing a Child's Fixed I/O Resources with the Parent 63
Addressing a Child's I/O in the Parent 63
Splitting a Child's Fixed I/O Resources 65
Addressing Generalized Data Transfer Registers in a Parent and Child 67
A GDT Example 67
Addressing A120 I/O Expansion Modules 70
Accessing A120 I/O LOCs 70
How the I/O Map Editor Displays an A120 I/O LOC 70
Checking the Number of Points Used for A120 I/O 71
Monitoring the Communications/Health Status of the PLC 72
Accessing the Comms/Health Displays 72
Monitoring Health Status of the I/O Locations 72
Monitoring the Comm Error Counters 74

Chapter 5 Ladder Logic Programming 75
Logic Edit Mode 76
Accessing Logic Edit Mode 76
Cursor Movement in Logic Edit Mode 76
Moving within a Network 77
Moving between Networks and Segments in Logic 77
Creating a New Network 78
Toggling between Logic Edit and Logic Data Modes 79
The Logic and Zoom Displays 80
Node Symbols in the Logic Display 80
Entering Coils, Contacts, and Shorts with an HHP 82
Legal Entry Locations 82
Entering a Logic Element in a Network 83
Entering a Vertical Short 84
How the HHP Displays Contacts, Coils, and Shorts 85
Displaying Null Nodes 86
An Example Network 86
Entering Ladder Logic Instructions with an HHP 88
Legal Entry Locations 88
Placing an Instruction in a Network 89
Overwriting an Old Instruction with an New One 91

How the HHP Displays Ladder Logic Instructions 92
A One-high Instruction 92
A Two-high Instruction 92
A Three-high Instruction 93
Errors of Range and Type 95
Range Errors 95
Data Type Errors 95
Special Commands for Editing Logic 96
Command Descriptions 96
How to LMARK a Discrete 96
Opening and Closing Rows in a Network 97
Opening and Closing Columns in a Network 98
Deleting a Network from the Ladder Logic Program 99
Checking for Available User Logic Memory 100
Escaping from the Command List 101
Searching Nodes for Program Data 102
Starting a Search 102
Ending a Search 102
Searching for a Reference Number 102
Special Search Capabilities for Discrete References 104
Tracing a Coil 107
Finding a Subroutine 107

Chapter 6 Monitoring and Editing Data Reference Values 109
Logic Data Mode 110
Accessing Logic Data Mode 110
Entering Reference Data 110
How to Get Reference Data from the Logic Edit Screen 111
Reference Formats and Entry Fields 112
Discrete Formats 112
Register Formats 113
Moving through the Logic Data Screen 115
Moving between Line Entries 115
The < **next** > and < **prev** > Keys 115
Editing Discrete Reference Data 117
Editing Register Reference Data 118
Selecting a Format 118
Editing Decimal and Hexadecimal Data 118
Editing ASCII Data 119
Editing Binary Data 120

Chapter 7 Using the HHP as a Data Transfer Device 121
Modes Of Data Transfer 122
Transfer/Save Mode Options 122
Slave HHP Mode Options 122
How to Access Transfer/Save Mode 122
How to Access Slave HHP Mode 123
Some Important Cross-transfer Considerations 124
Compatibility between Micro PLC Models 124
Editing the Target I/O Map after a Data Transfer 124
Transferring Data Created on another Panel to the HHP 124
A PLC-to-HHP Data Transfer 126
Accessing the PLC-to-HHP Transfer Screens 126
Initiating the Data Transfer 126
An HHP-to-PLC Data Transfer 128
Accessing the PLC-to-HHP Transfer Screens 128
Initiating the Data Transfer 128
Comparing HHP and PLC Data 130
Accessing the HHP-PLC Verify Screens 130
Initiating the Program Verification 130
What to Do if You Find an Unexpected Miscompare 131
Saving a Logic Program to PLC Flash 132
Accessing the Save-to-Flash Screens 132
Selecting a Start-up State for the PLC 132
Initiating the Save 133
Hardware Requirements for Slave Mode Operations 134
Uploading and Downloading Logic with an HHP in Slave Mode 136
Powering Up the HHP 136
Accessing Computer Transfer Mode 136
Selecting a PLC Model Number for the Slave 137
Uploading a Program to the Computer 137
Downloading a Program to the HHP Slave 138
Changing HHP Executive Firmware 141
Updating the Executive with a Loader Utility Program 141
Accessing Modfax 141
Accessing the Customer Service Bulletin Board 141
HHP Executive Download Procedures 142

Chapter 8 Using an HHP as a Simple Messaging Display 145
The HHP's Simple Messaging Mode 146
Accessing the Simple Messaging Screen 146
ASCII Characters Supported by the HHP 147
Using an HHP as a Message Display 148
A Ladder Logic Example: Displaying the ASCII Character Set on the HHP 149
Using an HHP as an Message Response Tool 151
A Ladder Logic Example: Responding to the PLC 151

Chapter 9 The HHP's Optional Password Security System 155
The Four Levels of Password Security 156
Accessing a Security Level without a Password 156
Level 1 Security Privileges 156
Level 2 Security Privileges 156
Level 3 Security Privileges 157
Admin Security Privileges 157
Password Construction 157
Accessing the Security Scheme 158
Monitoring Security Status without Admin Privileges 158
Becoming Admin 158
Enabling Password Security 160
Disabling Keyswitch Security 161
Assigning Passwords 162
Password Hierarchy 162
Creating a Level 3 Password 162
Creating a Level 2 Password 163
Customizing the Admin Password 163
Deleting a Password 164

Appendix HHP System Messages 165

Modicon Hand-held Programmer User Guide

Modicon Hand-held Programmer User Guide

Chapter 1
Introduction to the Hand-held Programmer

- About this Book
- A Panel Hardware Overview
- Screen Navigation and Keystroke Dynamics
- Providing Power to the HHP
- Hardware Specifications

About this Book

Organization

This book provides the necessary instructions for understanding the various operating modes of the 520VPU19200 English Language Hand-held Programmer (HHP).

- Chapter 1 describes the mechanical and electrical characteristics of the HHP panel and the ways to provide the HHP with operating power
- Chapter 2 describes power-up and start/stop procedures for the Micro PLCs
- Chapter 3 describes how to establish a valid configuration for a Micro PLC
- Chapter 4 describes how to address physical inputs and outputs (I/O) in your application to locations in the PLC's memory
- Chapter 5 describes how ladder logic programs can be created, monitored, and displayed on an HHP
- Chapter 6 describes the data reference editing and monitoring capabilities of the HHP
- Chapter 7 describes how to transfer logic programs between devices using the HHP; it also describes how to upgrade the executive firmware in an HHP
- Chapter 8 describes how to use the HHP as a simple messaging display panel for text strings generated by the PLC logic program
- Chapter 9 describes the optional password security available in an HHP
- An appendix listing HHP system messages is also included

Product Capabilities

The HHP is designed as a low-cost programming panel for the Modicon Micro PLCs. It supports all CPU311, CPU411, CPU512, and CPU612 Micro PLC models operating under standard ladder logic firmware.

The HHP can be used to:

- Start and stop the PLC
- Monitor the PLC's status
- Configure the PLC
- Address the PLC's fixed and expanded I/O points to locations in its memory
- Transfer programs to and from other Micro PLCs and other programming panels—e.g., Modsoft Lite
- Program and edit ladder logic
- View power flow in ladder logic while the PLC is running
- Edit and monitor PLC data references
- Initiate memory saving operations in the PLC
- Act as a simple messaging display device, showing ASCII character strings generated by the Micro PLC

A Panel Hardware Overview

First, let's take a look at the physical aspects of the HHP panel. It contains a LCD screen that displays character data 20 columns across by four rows down, a multipurpose LED indicator, and a keypad with 29 dual-function keys.

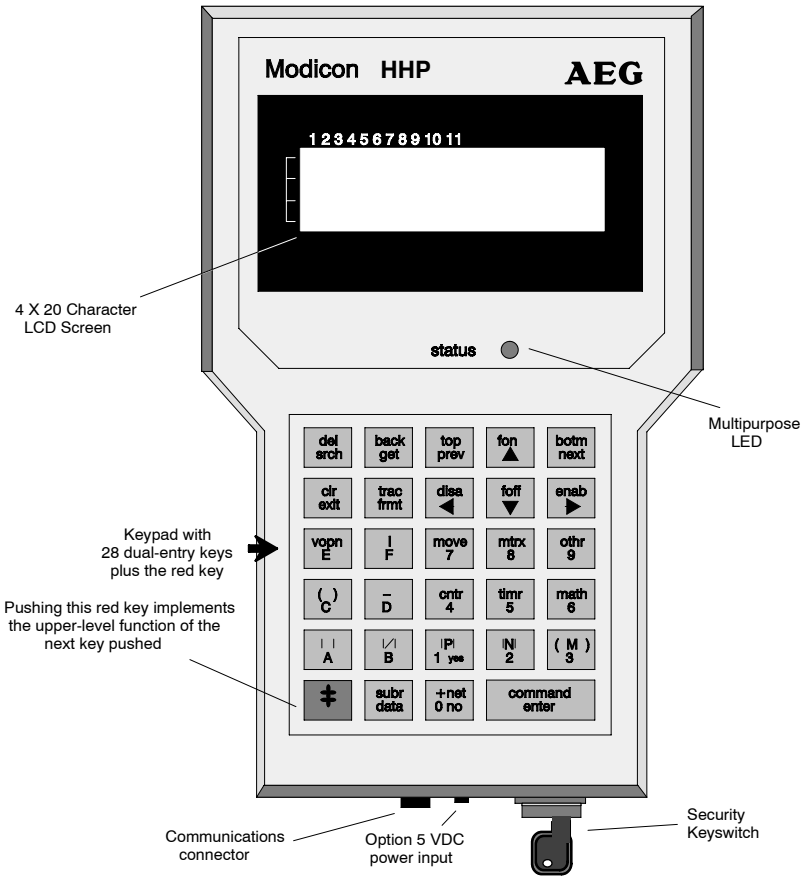
The Liquid Crystal Display

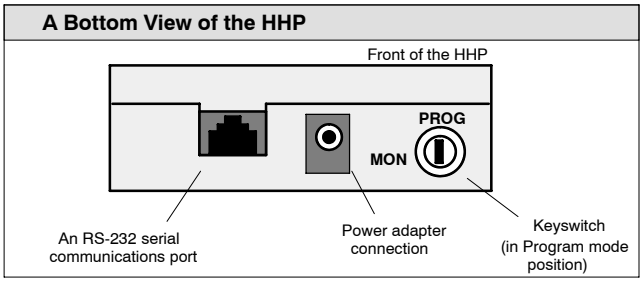
The first 11 columns of the LCD are labeled above the screen, and the four rows are marked to the left of the screen—these markings are provided to help clarify ladder logic displays when they are on the screen.

The Multipurpose LED

Below the LCD is a red light-emitting diode (LED). It may go ON or flash for a variety of reasons, depending on how the HHP is being used:

- It flashes when the HHP is receiving power but has failed its start-up diagnostics (For a discussion of start-up procedures, see Chapter 2)
- It is ON when the HHP is being downloaded with a user logic program or with new executive firmware (for a discussion of downloading procedures, see Chapter 7)





- It is ON when the HHP senses an LMARKed coil in the ON state running under the PLC's control (see pages 94 ... 95 for LMARK details)
- It is ON when the HHP in ASCII display mode receives an incoming ASCII message that contains the ASCII bell character (007); the LED can be turned OFF by pressing any key (see page 144 for a discussion of the bell character)

Communication and Power Connections

The RS-232 serial port on the HHP allows the panel to make a communications connection with a Micro PLC or with a computer.

When connected to a Micro PLC, the HHP receives its power from the PLC via the interconnecting communications cable. The HHP draws 150 mA from the PLC.

When connected to a personal computer, the HHP must draw its power from an external power supply. An optional AC-to-DC converter kit is available from Modicon for this purpose. The wire from the AC converter plugs into the power adapter connection (shown above) on the bottom of the HHP.

More details on the HHP power connections are given on pages 11 and 12.

The Security Keyswitch

The keyswitch is one of two security systems available on the HHP. (The other system, a four-level password security scheme—is described in detail in Chapter 9.) The HHP can be put in program (PROG) mode or monitor (MON) mode via this two-position keyswitch, located on the bottom right of the panel.

Program Mode

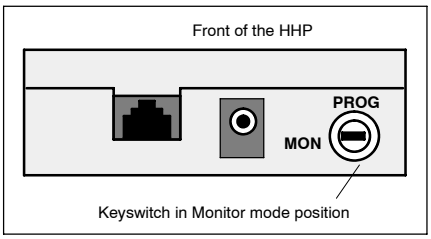
In program mode, the HHP allows you to auto-login to a Micro PLC and gives you access to all the functions available in the HHP, including editing logic and data in the PLC.

The keyswitch position for program mode is with the flat face of the key facing perpendicular to the front of the HHP (see the illustration above).

Monitor Mode

In monitor mode, the HHP cannot stop/start a PLC or make any changes to the PLC's state memory or user logic; power flow in the logic networks is not displayed. Only the transfer of user logic into the HHP (including verification) is allowed; you cannot load user logic to a PLC or save to flash with an HHP in monitor mode.

The keyswitch position for monitor mode is with the flat face of the key parallel to the front of the HHP, as shown in the following illustration.

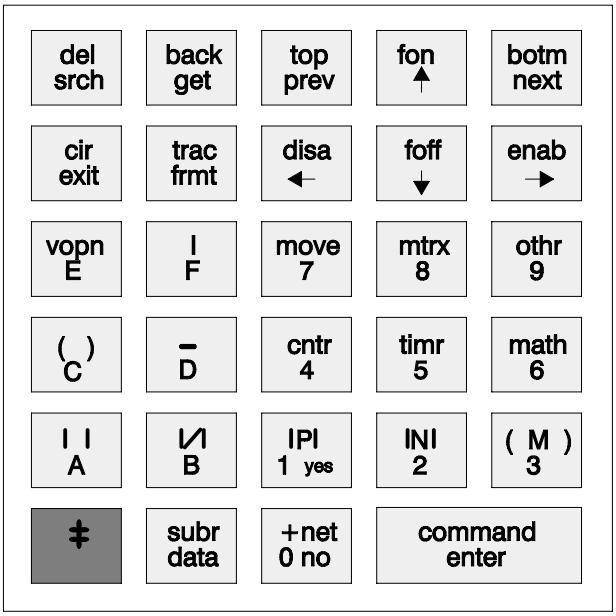


The Keypad

The keypad layout comprises 29 keys, 28 of which have dual functions. The red **command enter** key in the bottom left corner of the keypad is used to shift the sense of the next key to be entered. When a key is pushed by itself, the command or character inscribed on the lower portion of the key will be executed. When you push the red key before pushing one of the other keys, the command or character inscribed (in red) on the upper portion of the key will be executed.

For example, pushing the A key inserts the lower-level instruction—the alphabetical character or hexidecimal number A—on the screen. Pushing the key sequence ‡ + A inserts the upper-level instruction—a normal contact—in a logic network display.

The commands and characters associated with each key will be described throughout the book in the context of the various HHP operating modes.



Screen Navigation and Keystroke Dynamics

Moving through Data Fields in an HHP Screen

You can navigate through various fields in an individual screen by moving the cursor with the fon and foff arrow keys. The HHP allows you to move only into those fields where you can take a user action—the cursor cannot be moved into a field that displays only status information.

For example, the screen below shows a top-level HHP screen with four menu items on the left and several status items on the right:

Operations	Rev 1.00
Map I/O	SINGLE
Xfer/save	STOPPED
Slave hhp	CPU31100

By pushing fon or foff, you can move up or down through the list of four menu choices—**Operations**, **Map I/O**, **Xfer/save**, and **Slave hhp**. You cannot move the cursor into the fields on the right side of the screen.

Selecting Items from a Scrolled List in a Field

Often you need to select a setting from a list of possible choices as you configure and program your PLC (or as you set up a cross-transfer of programs from one model to another). Only one of the possible entries appears in a field on the screen at a time, so you need to scroll through the list using the disa and enab arrow keys.

For example, if you are writing a ladder logic program with the HHP and you want to create a **DIV** (divide) instruction in a logic network, you do so by pushing the ‡ math 6 keys while the HHP is in the logic editor. This key sequence accesses the math instructions. There are six possible selections that you can make—the **ADD**, **SUB**, **MUL**, **DIV**, or **EMTH** instruction or **CANCEL** to exit from the **math** menu without inserting an instruction.

Using enab, you can scroll forward through the list of **math** instructions.

Using disa, you can scroll backward through the list. Each time you push one of these arrow keys, the name of the instruction changes in the zoom display. Keep pushing an arrow key until **DIV** appears.

Entering Numeric Data in a Field

When numerical data is required in a data field on an HHP screen, you need to enter that data by pushing number keys (or the alphabetical keys for some hex numbers) on the keypad. For example, if you are configuring a parent PLC, you need to enter the number of child PLCs that the parent can access on the I/O expansion link. That number can range from **1** ... **4**.

☞ **Note** An exception is entry of baud rates when you set a custom configuration. Baud choices can differ depending on the comm port you are using and are not intuitive entries. They can be selected by scrolling with the disa ◄ and enab ► arrow keys.

When you enter a string of multiple numeric characters into a field—e.g., in editing in logic data mode—the field is always filled from the right as you enter characters.

Moving between HHP Screens

You can move from one screen to another in several ways:

- Select a menu item on the current screen by placing the cursor on it and pushing command enter
- Confirm a data setting displayed on a screen by pushing command enter
- Move from one screen in a series to the next screen by pushing botm next or move from one screen in a series to the previous screen by pushing top prev
- Move to the top of a series of screens or exit from an editor mode in the HHP by pushing clr exit

When a screen that is one in a series appears on the screen, a right-pointing arrow (→) appears in the top right corner of the screen. This arrow indicates that there are more screens that you can page through by pushing botm next .

The arrow does not appear in the last screen in the series.

The top-level entry screens, for example, comprise a two-screen series, where the first screen has an arrow in the upper right corner and the second screen does not:

```
Logic edit  | RTU   1   →
Logic data  |  SINGLE
Plc config  |  STOPPED
Next menu   |  CPU311 00
```

botm next

```
Operations  | Rev 1.00
Map I/O     | SINGLE
Xfer/save   | STOPPED
Slave hhp   | CPU31100
```

If you push botm next when you reach the last screen in the series, it wraps back to the first screen in the series. In some multi-screen series, the top prev key can be used to move sequentially backward through the series.

The Look of the Cursor

The cursor can be displayed in any one of three possible ways:

- As a single character underline that can be placed under the first letter of a text selection or under a ladder logic node symbol on the HHP screen
- As a blackened rectangle flashes over a selected character or node symbol on the HHP screen
- A combination of the above two, where the cursor alternates between the flashing blackened rectangle and the underline

The style of cursor you want to use with your display can be selected as you start up the PLC with the Hand-held Programmer. (For details, see page 17.)

Providing Power to the HHP

The HHP needs 150 mA from an external power source in order to operate. If the HHP is connected to a Micro PLC, it draws its power directly from the PLC. If it connected to another device—e.g., a personal computer—it must draw its power from an independent external power source.

Connecting to a Micro PLC

When a communication connection is made between the HHP and a PLC, the HHP automatically receives power from the PLC. Power is transmitted over the RS-232 communications cable that connects the two devices.

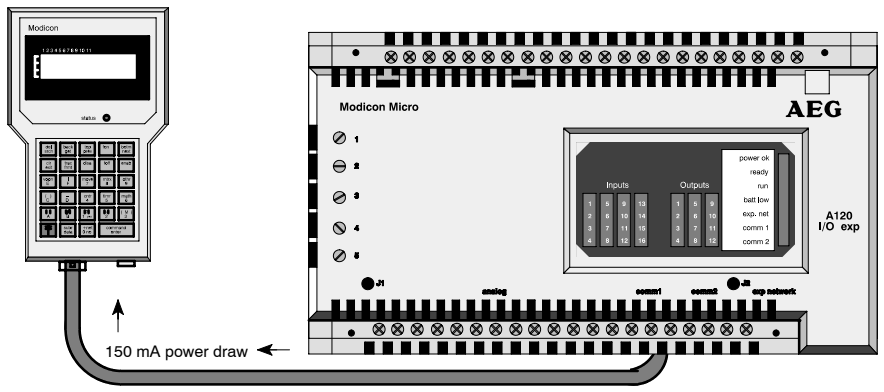
Use one of the following RS-232 cables to connect the HHP to the PLC:

RS-232 Communications Cables	
Length	**Part Number**
1 m (3 ft)	110XCA28201
3 m (10 ft)	110XCA28202
6 m (20 ft)	110XCA28203

Each of these three cables is an eight-position, foil-shielded, flat telephone cable with male 8-pin RJ45 connectors on each end. Plug one RJ45 connector into the HHP's communications port and the other one into an RS-232 serial port on a Modicon Micro PLC.

As soon as the HHP receives power, it begins running its power-up diagnostics and attempts to establish communications with the PLC.

An HHP-PLC Connection



RS-232 communications cable

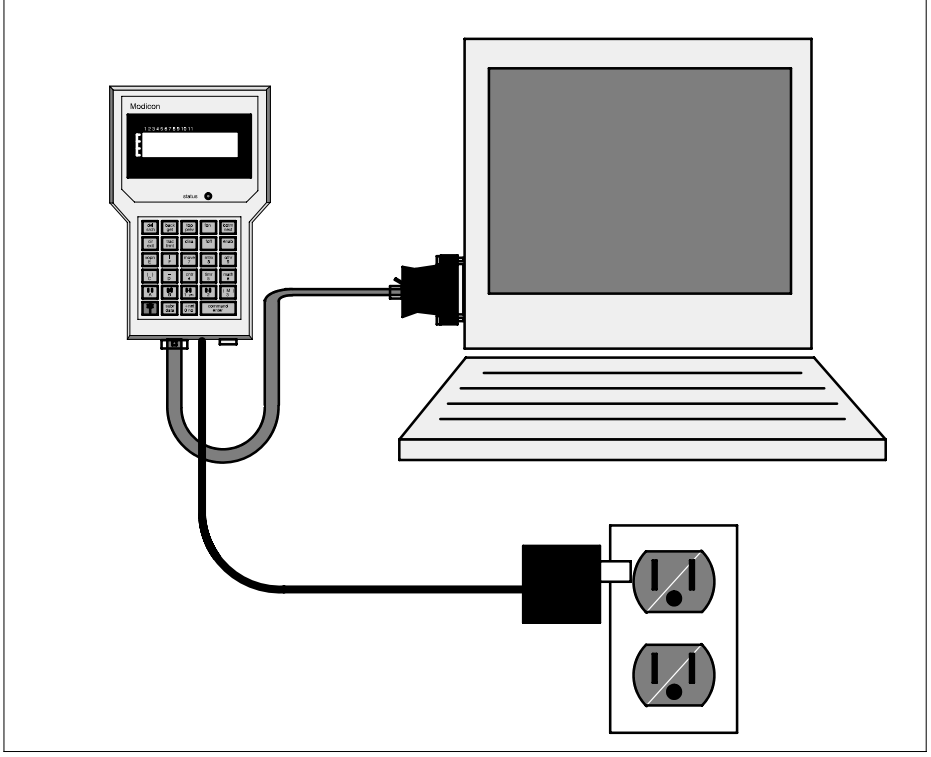
Using an Independent Power Supply

There may be times when you need to connect the HHP to a device other than a PLC. For example, to download new executive firmware or upload a logic program to the HHP, you may need to connect the HHP to a personal computer.

When the HHP is connected to a non-PLC device, it must draw its power from an independent DC source.

Modicon offers an 115 VAC-to-5 VDC power converter for this purpose as part of an optional HHP Computer Transfer Mode Kit (110VIA19200).

Connecting an HHP to an Independent DC Power Source



Hardware Specifications

Environmental

Operating temperature	0 ... 50 degrees C
Storage temperature	–20 ... 70 degrees C
Relative humidity	95% noncondensing
Altitude	3800 m (15,000 ft)
Shock	30 g for 11 ms, 3 pulses/axis for up to 18 pulses
Vibration	10 ... 62 Hz @ .075 mm displacement amplitude, 62 ... 500 Hz @ 1 g
Dimensions	
Height	165 mm (6.5 in) with no key inserted 196.75 mm (7.75 in) with key inserted
Width	107 mm (4.21 in)
Depth	41.07 mm (1.62 in) including mounting clips
Weight	.27 kg (.6 lb)
Agency approvals	Designed to meet UL, CSA, and VDE requirements for industrial and process control equipment

Electrical

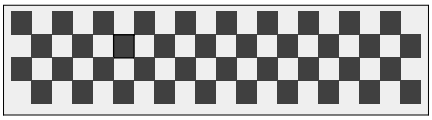
Maximum power draw	150 mA
Electrical immunity	
ESD	IEC 801-2, level 3
Radiated EMI	IEC 801-3, level 3
Fast transient	IEC 801-4, level 2

Chapter 2
Getting Started

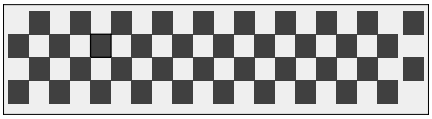
- Powering Up the HHP
- Configured and Unconfigured Micro PLCs
- Modes of Communication with a Configured PLC
- Operating the PLC

Powering Up the HHP

As soon as power is applied to the Hand-held Programmer, the LED flashes, and then liquid crystal display shows two checker-board patterns:



followed by



It is while these two screens are flashing that you may select your cursor display style (see page 19).

The screen clears briefly while the HHP performs its power-up diagnostics. Then the HHP tries to establish communications with the device connected to its RS-232 port, displaying the following message:

```
Hunting for Comms
Please  wait . . .
```

Communicating with a PLC

If the HHP is powered by a Modicon Micro PLC, one of three possible screens will appear on the LCD as soon as communication is established:

❒ If the PLC is already configured:

Logic Edit	RTU 1 →
Logic Data	SINGLE
Plc Config	STOPPED
Next Menu	CPU31100

❒ If the PLC is unconfigured:

Plc Config	RTU 1
Slave Hhp	NO CONFIG
	CPU31100

❒ If the PLC cannot be configured:

```
          PLC
   IN KERNEL MODE
```

In the first two cases, notice that the LCD divides into two display areas. The display on the left lists a set of menu picks, and the display on the right provides status information about the PLC.

Menu Displays

The cursor operates only in the menu display on left side of the display. Using the menu picks, you can navigate through the levels of HHP operation. The various paths that may be followed using the menu selections are discussed in detail throughout this book.

To make a menu selection, move the cursor under the first character of a menu pick, then push command enter .

The cursor can be moved up or down through the menu selections with the fon ▲ and foff ▼ keys. Both keys will perform a wrap-around on the menu list—e.g., if the cursor is on **Next Menu** in the first screen and you push the down arrow key, the cursor will appear on the **Logic Edit** menu pick.

Status Displays

The PLC status display on the right of the LCD is for information purposes only —it cannot be edited.

The top line of the status display tells you about the comm port of the PLC—the communications protocol it is using (**RTU** or **ASCII**), and its Modbus address (**1** ... **247**).

The second status line tells you the operating mode in which the PLC is configured (**SINGLE**, **PARENT**, **CHILD**, or **NO CONFIG**).

The third status line tells you the PLC's current operating state (**RUNNING** or **STOPPED**). The fourth status line tells you the model number of the Micro PLC.

☞ **Note** In the case of the third screen, when your PLC is in *kernel mode* , no further interaction is possible between the HHP and the PLC. New executive firmware must be downloaded to the PLC.

If the HHP Cannot Establish Communication

The HHP is able to establish valid communications only with a Modicon Micro PLC. Upon power-up, the HHP always attempts to establish that communication even when a PLC is not connected to it.

If communications fail to be established, the following **NO COMMS** screen appears:

Hhp port	NO COMMS
Slave Hhp	SINGLE
	UNKN PLC

Why Communications Cannot Be Established

The **NO COMMS** screen appears on the LCD if:

- You plug a DC power supply into the HHP's power adapter connection but do not connect another device to the HHP's RS-232 communications port
- You plug a DC power supply into the HHP's power adapter connection and connect a non-PLC device—e.g., a personal computer—to the HHP's RS-232 communications port
- You connect a Micro PLC to the HHP's RS-232 communications port, but some communications error occurs in the connection

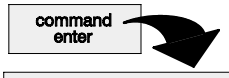
Potential HHP-PLC communication errors could be caused by:

- Comm port parameters in the PLC that do not match the comm port parameters of the HHP
- A Modbus network address in the PLC that is not recognized by the HHP
- A possible hardware problem in one of the two devices

Establishing Comms when a PLC's Modbus Address Is Not 1

The Modbus address of the PLC may range from 1 ... 247; 1 is the most commonly used address, but it is not always the correct address. As it attempts to establish communications with a PLC, the HHP looks for a Modbus address of 1 on the PLC. If the PLC has a Modbus address other than 1, the **NO COMMS** screen appears. Proceed as follows:

Hhp port	NO COMMS
Slave Hhp	SINGLE
	UNKN PLC



Cpu address =			001
Comm parms =			A
BAUD	MOD	STP	PAR
9600	RTU	1	EV

- Enter the correct PLC address under the cursor on the top line. If the comm port parameters are the same as those shown on the bottom line of the screen—e.g., 9600 baud/RTU with 1

stop bit and even parity—push command enter to confirm the address entry, then push clr exit to leave the screen and hunt for comms. When the HHP establishes communications with the PLC at the specified Modbus address, it returns the standard top-level HHP menu screen.

If the comm port parameters are different from those shown on the bottom line of the screen, enter the the desired Modbus address on the top line and push command enter to confirm the number. Then push foff ▼ to move the cursor under the **A** or **B**, then push enab ▶ to toggle the **A** to **B**. or the **B** to **A**.

☞ **Note** Only two sets of comm parameters can be set by the HHP for its **comm1** port. An **A** on the second line indicates 9600 baud/RTU with 1 stop bit and even parity, and a **B** on the second line indicates 2400 baud/ASCII with 1 stop bit and even parity.

When you have specified the desired Modbus address for the PLC and the correct comm parameters for communicating with the HHP port, push clr exit to leave the screen and hunt for comms. When the HHP establishes communications with the PLC at the specified Modbus address with the correct comm port parameters, it returns the standard top-level HHP menu screen.

Establishing Comms when a PLC's Modbus Address Is Unknown

There may be times when the HHP fails to establish communications with a PLC whose Modbus address is unknown to you. In this case, make sure that the HHP is connected to the **comm1** port of the PLC.

Using the **NO COMMS** screen, enter an address value of **000**:

```
Cpu address =      000
Comm parms  =      A
 BAUD   MOD   STP     PAR
 9600   RTU    1       EV
```

Push command enter to confirm the address entry, then clr exit to leave the screen and hunt for comms.

The HHP systematically tries all combinations of addresses 1 ... 247 with both the **A** and **B** type port parameters. The hunt for comms could take on the order of ten minutes, worst-case. When the HHP establishes communications with the PLC at the specified Modbus address with the correct comm port parameters, it returns the standard top-level HHP menu screen.

The Slave Hhp Menu

Slave Hhp can be selected if you are connected to Micro PLC and you want to advance through the security passwords or if you are connected to a computer and you want to download new HHP executive firmware from it.

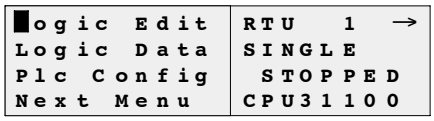
If the HHP Fails its Start-up Diagnostics

The HHP will fail its start-up diagnostics if its executive firmware is determined to be invalid. In this case, the HHP goes into its *kernel mode* and the LCD remains blank or displays some meaningless pattern such as checks or bars.

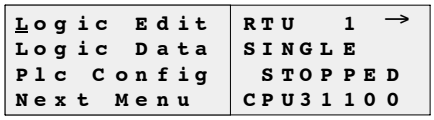
Selecting a Cursor Display

You can select any one of three possible cursors on the HHP liquid crystal display:

- *Option 1*—a blackened rectangle that flashes on the first character in a selected text entry or on a ladder logic node symbol on the HHP screen:



- *Option 2*—the same blackened rectangle flashing alternating with an underlined character on a character or node symbol on the HHP screen
- *Option 3*—a single-character underline that appears constantly under a character or node symbol on the HHP screen:



To establish option 1 as your cursor style, hold down the

key while the HHP displays the checker patterns at power up.

To establish the option 2 as your cursor style, hold down the

key while the HHP displays the checker patterns at power up.

To establish the option 3 as your cursor style, hold down the

key while the HHP displays the checker patterns at power up.

Once a cursor style has been selected, it is written to the HHP flash and remains in effect each time you power up the HHP with the PLC. You do not need to re-select it each time you power up the HHP.

If you never select a cursor style during the HHP power up, option 3—the underline—becomes the default cursor style. The underline cursor is shown in screen examples throughout this book.

Configured and Unconfigured Micro PLCs

If the HHP receives power from a Modicon Micro PLC that is unconfigured, a set-up screen similar to the one below appears in the LCD:

plc Config	RTU 01
Slave hhp	NO CONFIG
	CPU31100

The PLC is *unconfigured* when it is being powered up for the first time or when it has had all its previous memory contents cleared.

Configuration Modes

The PLC must be configured before you can program or operate it. A *configuration* is a table of parameter values that define the PLC's operating parameters and conditions.

You can set up this parameter table with an HHP using the *Auto config* mode and customize the table using the *Set config* mode.

Autoconfig Mode

The PLC can automatically establish a configuration table for itself based on the model of Micro PLC you are using and on its operating mode—single, parent, or child. The autoconfiguration process sets up valid configuration parameters for the fixed I/O and communication resources of the unit and reserves memory space for handling additional resources—e.g., A120 I/O, child I/O resources to be accessed by a parent.

Autoconfig mode is a quick and efficient way to set up a single PLC that does not use A120 I/O expansion or a child PLC whose fixed resources are to be accessed by the parent. In both these cases, the autoconfiguration parameters (see Chapter 3) give you everything you need to operate the PLC.

Autoconfig mode gives you only a preliminary configuration for a parent PLC, a single PLC that uses A120 I/O, or a child that uses A120 I/O or any of its own fixed resources. In these cases, you will need to edit the autoconfig parameters in order to finalize the configuration.

Set Config Mode

Whenever you need to set up operating parameters for I/O or communication resources apart from those fixed on your PLC, you need to use set config mode.

With an HHP in set config mode, you can modify:

- The communication parameters of the RS-232 port(s) on the PLC—**comm 1** and, in the 110CPU512 and 110CPU 612 models, **comm 2**
- The parameters and destination of the designated simple ASCII port on the PLC

View Config Mode

In view config mode, you can view:

- Your data reference assignments
- The number of segments in the logic program
- The amount of user logic available and already used
- The time-of-day clock registers
- The free-running timer register

For a detailed description of how to work in the set config and autoconfig modes, refer to Chapter 3 of this book.

Modes of Communication with a Configured PLC

Let's look more closely at the screen that appears in the LCD when you establish communications with a configured PLC:

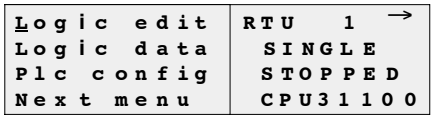
From this top-level screen you can get to virtually all the other PLC-related operations available in the HHP.

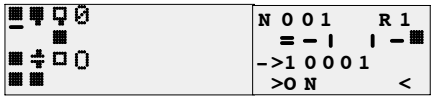
Top-level PLC Operating Modes

Selecting one of the first three menu picks listed on the left side of the screen takes you into one of the following operating modes:

Logic Edit Capabilities

Selecting **Logic edit** from the top-level menu sends you to a series of screens where you can create, edit, and/or monitor a ladder logic program. Here is the screen/command sequence for accessing the logic edit mode:





You enter a display of the first network of ladder logic. (See Chapter 5 for all operational details regarding logic edit mode.)

Logic Data Capabilities

Selecting **Logic data** from the top-level menu sends you to a screen where you can edit and/or monitor reference data. Here is the screen/command sequence for accessing the logic data mode:

When you enter a 0*x*, 1*x*, 3*x*, or 4*x* reference number in a data field to the left of an = , the current state or content of that reference is displayed in the field to the right of the = .

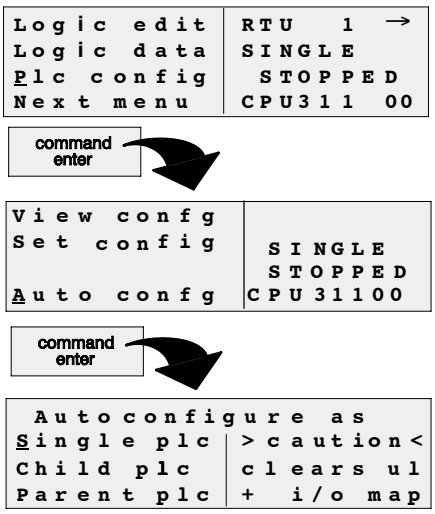
Note You can also access logic data mode from the logic edit mode (and toggle back and forth between the two modes) by pushing subr data .

See Chapter 6 for a detailed discussion of HHP operations in logic data mode.

Config and Autoconfig Capabilities

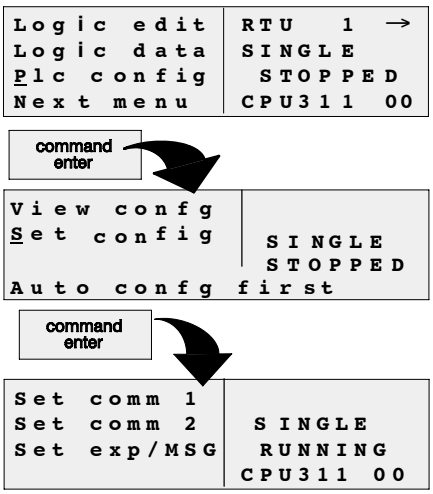
Selecting **Plc config** from the top-level menu of the HHP sends you to a series screens where you can choose to automatically configure or customize the current configuration of your Micro PLC.

Here is the screen/command sequence for autoconfiguring the PLC:



See pages 30 ... 33 for a detailed discussion of how to autoconfigure a Micro PLC with the HHP.

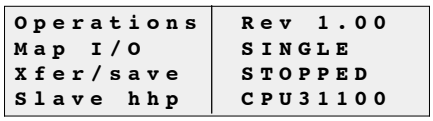
Here is the screen/command sequence for customizing the configuration of the PLC:



See pages 34 ... 39 for a detailed discussion of how to customize a config.

Moving to the Next Top-level Menu

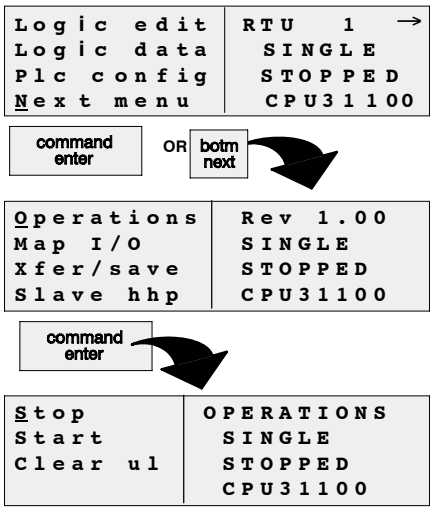
There is a second top-level menu screen, as indicated by the → in the top right corner of the screen. To reach this next menu, either push botm next or move the cursor under **Next Menu** and push command enter :



The second top-level screen shows the current revision level of the PLC executive on the top right side of the screen.

PLC Operations

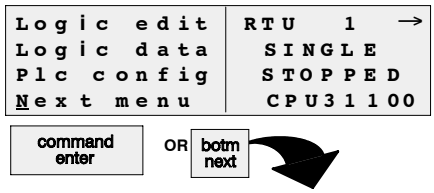
Selecting **Operations** from the second top-level menu sends you to a screen where you can stop and start the PLC and you can clear the current user logic program from the PLC's user memory.



See page 24 for a discussion of **Operations** functions.

I/O Map Capabilities

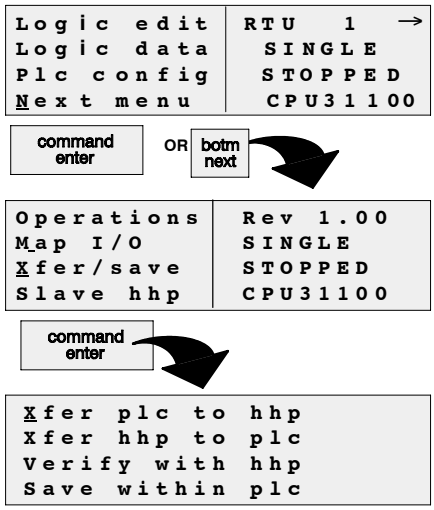
Selecting **Map I/O** from the menu on the second top-level HHP screen sends you to the HHP's I/O map editor.



See Chapter 3 for a detailed discussion of how to I/O map a Micro PLC with the HHP.

Program Transfer Mode

The **Xfer/save** mode can be accessed in an HHP via the following menu path:



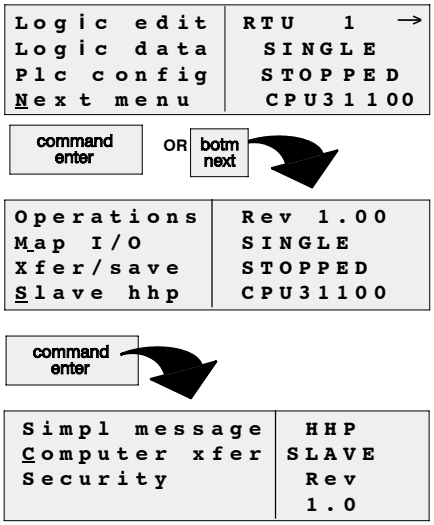
Selecting **Xfer/Save** sends you to a series of screens where you can:

- Transfer the contents of the PLC to the HHP
- Transfer the contents of the HHP to the PLC
- Save the current contents of the PLC to its Flash RAM

See Chapter 7 for a detailed discussion of how to use the HHP for Computer Transfer Mode and Save to Flash operations.

HHP Slave Mode Capabilities

Selecting **Slave hhp** sends you to a series of screens where you can communicate with a non-PLC device or make the HHP a slave to the PLC.



HHP slave mode can be invoked to load new HHP executive firmware to the HHP's Flash RAM, to copy user logic to the HHP from an MMI device, or to implement simple messaging.

To return to the first top-level HHP screen without selecting any of the menu picks on the second screen, push clr exit .

Operating the PLC

Once the PLC has been configured, it can be started—i.e., put in *run* mode—and stopped using the HHP's **Operations** screens. In HHP **Operations** mode, you can also clear existing ladder logic from the PLC memory.

Accessing the Operations Screens

The **Operations** screens can be accessed from the second top-level HHP screen using a screen/command sequence similar to this:

```
Logic edit   | RTU   1   →
Logic data   |  SINGLE
Plc config   |  STOPPED
Next menu    |  CPU31100
```

command enter OR botm next

```
Operations   |  Rev 1.00
Map I/O      |  SINGLE
Xfer/save    |  STOPPED
Slave hhp    |  CPU31100
```

command enter

```
Stop         | OPERATIONS
Start        |  SINGLE
Clear ul     |  STOPPED
             |  CPU31100
```

Starting a Stopped PLC

In the example above, we see that the PLC is currently stopped when **Operations** mode is entered. The cursor is initially positioned under the top left menu choice—**Stop**.

To start the PLC, simply push foff ▼ once to move the cursor under **Start** and push command enter .

The following confirmation screen appears:

```
        OPERATIONS
 start
 Reqest
 Confirm
```

Push command enter to put the PLC in run mode. The PLC starts running and the HHP returns to the top-level screen. The third line of the status information on the right side of the screen shows that the PLC is now **RUNNING**:

```
Operations   |  Rev 1.00
Map I/O      |  SINGLE
Xfer/save    |  RUNNING
Slave hhp    |  CPU31100
```

If you decide that you want to cancel the command and leave the PLC stopped when you reach the confirmation screen, push clr exit . The PLC remains stopped and the HHP returns you to the first **Operations** screen.

Stopping a Running PLC

To stop a PLC that is in run mode, go to the first **Operations** screen and, with the cursor positioned under **Stop**, push command enter .

The following confirmation screen appears:

```
        OPERATIONS
 stop
 Reqest
 Confirm
```

Push command enter to stop the PLC. The PLC stops running and the HHP returns to the top-level screen. The third

line of the status information on the right side of the screen shows that the PLC is now **STOPPED**:

```
Operations    Rev 1.00
Map I/O       SINGLE
Xfer/save     STOPPED
Slave hhp     CPU31100
```

If you decide that you want to cancel the command and leave the PLC running when you reach the confirmation screen, push clr exit . The PLC remains stopped and the HHP returns you to the first **Operations** screen.

Clearing User Logic from PLC Memory

To clear user logic program from the PLC memory, the PLC must be stopped. With the PLC stopped, go to the first **Operations** screen and position the cursor under **Clear ul**, then push command enter .

If you attempt to clear user logic while the PLC is running, an error message appears on the bottom line of the screen:

```
Stop          OPERATIONS
Start           SINGLE
Clear ul        RUNNING
Controller running
```

In this case, follow the instructions above for stopping the PLC, then try **Clear ul** again.

With the PLC stopped, the following confirmation screen appears:

```
        OPERATIONS
 Clear user logic
 Reqest
 Confirm
```

Push command enter to erase all ladder logic from the PLC's user program memory. The HHP returns to the top-level screen with the following system message on the bottom line:

```
Operations    Rev 1.00
Map I/O       SINGLE
Xfer/save     STOPPED
User logic cleared
```

Chapter 3
Configuring a Micro PLC

- The PLC Config Options
- How to Autoconfigure a Micro PLC
- Customizing a PLC Configuration
- Viewing Configuration Parameters

The PLC Config Options

If the PLC starts up in an unconfigured state, a top-level screen similar to the one below appears in the LCD of the Hand-held Programmer:

Plc Config	RTU 01
Slave hhp	NO CONFIG
	CPU31100

If the PLC already has been configured previously, the following top-level screen appears on the HHP:

Logic Edit	RTU 1 →
Logic Data	SINGLE
Plc Config	STOPPED
Next Menu	CPU31100

From either of these two the top-level HHP screens, you can access your PLC configuration options by selecting **Plc config**, then pushing command enter. The following screen then appears with three possible config options:

View confg	
Set config	SINGLE
	STOPPED
Auto confg	CPU31100

The Three Config Options

With the **View confg** selection, you can monitor the current PLC configuration parameters. You can only monitor the config parameters at this level, you cannot edit them. **View confg** capabilities are described on pages 41 ... 44.

With the **Set config** selection, you can edit some of the current PLC configuration parameters:

- PLC comm port parameters
- The PLC port used for simple messaging
- The I.D. number of a child PLC

You can also write the changes to the PLC with the **Set config** selection. **Set config** capabilities are described on pages 35 ... 40.

With the **Auto confg** option, you can assign a set of default configuration values to the PLC. **Auto confg** capabilities are described in detail on pages 30 ... 34.

Configuration Options for an Unconfigured PLC

If the PLC is not currently configured, you must **Auto config** it before you can execute **Set config** or **View confg** operations.

If you attempt to select **Set config** or **View confg** on a PLC that is unconfigured, the following message appears on the bottom line of the HHP screen:

View confg	
Set config	SINGLE
	STOPPED
Auto confg first	

Auto config is an extremely important step in the configuration process when you use the HHP to configure the PLC. Within the **Auto config** option resides the ability to:

- Specify an operating mode for the PLC
- Define the number of child PLCs on an I/O expansion link
- Define all the baseline parameters for your PLC configuration

The only other way to provide the PLC with these key configuration elements via the HHP is to download a valid PLC environment to the PLC from another source controller (see Chapter 7 for details).

Configuration Options for a Previously Configured PLC

Once the PLC has been provided with a valid configuration, the config parameters can be monitored with the **View confg** option and edited with the **Set config** option.

You can also run the **Auto confg** option on a PLC that has a valid configuration. You may want to do this to change the operating mode of the PLC—e.g., from a single to a parent.

Caution When the HHP writes new autoconfiguration parameters to a PLC that is already configured and programmed, it clears all previous I/O map and ladder logic data from system memory. This is true even if the new autoconfig parameters are the same as the old ones.

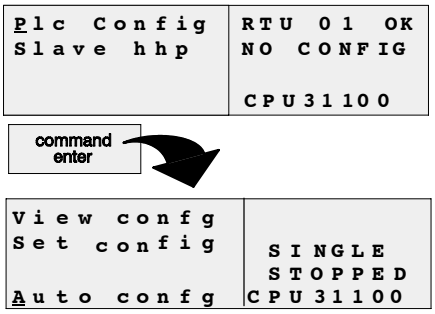
How to Autoconfigure a Micro PLC

When you autoconfigure a Micro PLC, you automatically assign a set of default configuration values to the PLC. The default assignments are based on the model type of the PLC and on the operating mode you specify—single, parent, or child.

If the PLC is not currently configured, you must autoconfigure it before you can operate it.

Accessing Autoconfig when the PLC Is Unconfigured

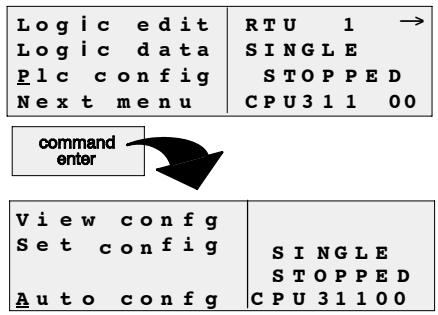
To access the autoconfig screens when the HHP is connected to an unconfigured PLC, move the cursor under **Plc config** and push command enter :



Then, depending on the operating mode you want to apply to your PLC, follow one of the three autoconfig procedures given on pages 31 ... 34.

Accessing Autoconfig when the PLC Is Configured

To access the autoconfig screens on the HHP, move the cursor under **Plc config** and push command enter :



Then, depending on the operating mode you want to apply to your PLC, follow one of the three autoconfig procedures given on pages 31 ... 34.

Autoconfiguring a PLC in Single Mode

If you want your PLC to operate as a single unit with the default configuration parameters of a single PLC, use the following screen/command sequence:

```
View confg      |
Set  config     |  SINGLE
                |  STOPPED
Auto confg      |CPU 31100
```

command enter

```
  Autoconfigure as
Single plc      |>caution<
Child  plc      |clears ul
Parent plc      |+ i/o map
```

command enter

```
Configure as SINGLE

<exit> to cancel
Begin to configure
```

Notice the caution on the right side of the middle screen above. If you want to escape this autoconfig screen without writing a new config (especially if you want to preserve an existing logic program in the PLC), push clr exit. The HHP returns you to the first of the three screens shown above, where you may select another option or escape out of the config mode.

To autoconfig the PLC as a single unit, push command enter with the cursor under **Begin to configure**. The HHP writes the single autoconfiguration parameters to the PLC and redisplays the top-level HHP screen. The newly autoconfigured PLC's operating mode is shown in the second status line on the right side of the screen:

```
Logic Edit      |RTU  1  →
Logic Data      |SINGLE
Plc Config      | STOPPED
Next Menu       |CPU31100
```

Autoconfig Parameters for a Single PLC

Parameter	110CPU Models	
	311 / 411	512 / 612
Number of 0*x* references	1024	1536
Number of 1*x* references	256	512
Number of 3*x* references	32	48
Number of 4*x* references	400	1872
Number of ladder logic segments	2 (the first for control logic and the second for subroutines)	2 (the first for control logic and the second for subroutines)
RS-232 port (**comm 1**)	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-232 port (**comm 2**)	N/A	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-485 port (**exp. link**)	Dedicated ASCII 8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit	Dedicated ASCII 8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit

Autoconfiguring a PLC in Child Mode

If you want your PLC to function as a child on an I/O expansion link with the default configuration parameters of a child PLC, use the following screen/ command sequence:

```
View confg      |
Set  config     |  SINGLE
                |  STOPPED
Auto confg      | CPU 31100
```

command enter

```
 Autoconfigure as
Single plc  | >caution<
Child plc   | clears ul
Parent plc  | + i/o map
```

command enter

```
Configure as    CHILD
Child no =          1
<exit> to cancel
Begin to configure
```

Initially the cursor appears on the second line of the screen under the number **1**. This number indicates the default child I.D number. You may leave the number set to **1** or change it to another number between **2** ... **4** by typing in the desired number from the HHP keypad.

☞ **Note** Each child PLC on the I/O expansion link must have a unique I.D. If another child has already been assigned I.D. number 1, then you must change the assignment to another integer.

Once the desired child I.D. number is set under the cursor, push foff ▼ . The cursor moves to the bottom line under **Begin to configure** .

If you want to escape this autoconfig screen without writing a new config, push cir exit .

If you want to write the default autoconfig parameters of a child to the PLC, push command enter .

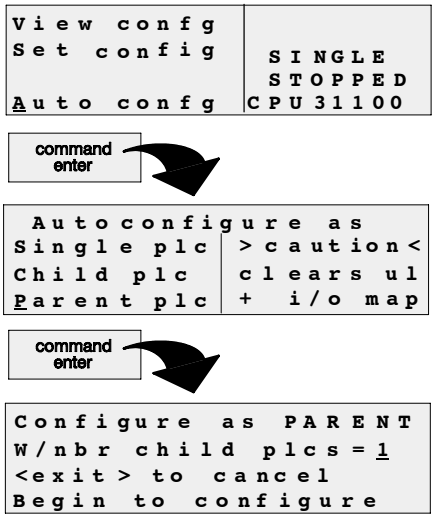
The HHP writes the child autoconfiguration parameters to the PLC and redisplays the top-level HHP screen. The newly autoconfigured PLC's operating mode is shown in the second status line on the right side of the screen:

```
Logic Edit      | RTU   1  →
Logic Data      | CHILD
Plc Config      |  STOPPED
Next Menu       | CPU31100
```

Autoconfig Parameters for a Child PLC		
	110CPU Models	
Parameter	311 / 411	512 / 612
Number of 0*x* references	1024	1536
Number of 1*x* references	256	512
Number of 3*x* references	32	48
Number of 4*x* references	400	1872
Child ID #	*must be user-specified*	*must be user-specified*
Number of ladder logic segments	2 (the first for control logic and the second for subroutines)	2 (the first for control logic and the second for subroutines)
RS-232 port (**comm 1**)	Modbus/ASCII toggling mode: 8-bit RTU/8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-232 port (**comm 2**)	N/A	Modbus/ASCII toggling mode: 8-bit RTU/8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-485 port (**exp. net**)	I/O expansion network: 9-bit data communications, 125 ,000 baud, 1 STOP bit	I/O expansion network: 9-bit data communications, 125 ,000 baud, 1 STOP bit

Autoconfiguring a PLC in Parent Mode

If you want your PLC to function as a parent on an I/O expansion link with the default configuration parameters of a parent PLC, use the following screen/ command sequence:



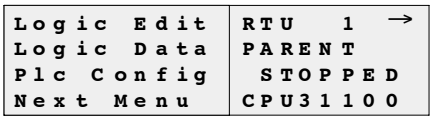
Initially the cursor appears on the second line of the screen under the number **1**. This number indicates how many child PLCs may be supported on the link. You may leave the number set to **1** or change it to another number between **2** ... **4**.

Once the desired number of child PLCs is set, push foff ▼ . The cursor moves to the bottom line under **Begin to configure** .

If you want to escape this autoconfig screen without writing a new config, push clr exit .

If you want to write the default autoconfig parameters of a child to the PLC, push command enter .

The HHP writes the parent autoconfig parameters to the PLC and redisplays the top-level HHP screen. The newly autoconfigured PLC's operating mode is shown in the second status line on the right side of the screen:



Autoconfig Parameters for a Parent PLC

Parameter	110CPU Models	
	311 / 411	**512 / 612**
Number of 0*x* references	1024	1536
Number of 1*x* references	256	512
Number of 3*x* references	32	48
Number of 4*x* references	400	1872
Number of child PLCs on the I/O expansion link	*must be user-specified*	*must be user-specified*
Number of ladder logic segments	2 (the first for control logic and the second for subroutines)	2 (the first for control logic and the second for subroutines)
RS-232 port (**comm 1**)	Modbus/ASCII toggling mode: 8-bit RTU/8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1	Dedicated Modbus mode: 8-bit RTU communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-232 port (**comm 2**)	N/A	Modbus/ASCII toggling mode: 8-bit RTU/8-bit ASCII communications, 9600 baud, even parity, 1 STOP bit, Modbus address #1
RS-485 port (**exp. net**)	I/O expansion network: 9-bit data communications, 125 ,000 baud, 1 STOP bit	I/O expansion network: 9-bit data communications, 125 ,000 baud, 1 STOP bit

Customizing a PLC Configuration

There may be times when you want to modify some existing configuration parameters to suit your application needs. In this case, you need to select the **Set config** option. **Set config** can be entered only when the PLC has a valid configuration.

Note If your PLC is currently unconfigured, you must must first autoconfigure the PLC in the desired operating mode before you can make the **Set config** selection.

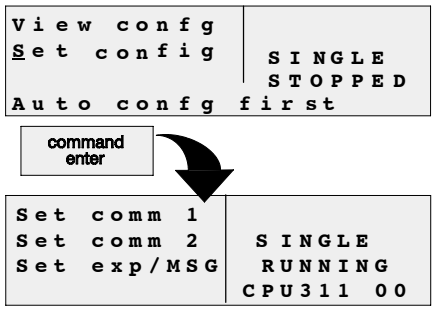
Config Parameters that Can Be Edited

The **Set config** screens allow you to:

- Change the communication parameters for all the ports on your PLC
- Change a port on the PLC that may be used for simple messaging
- Change the I.D. number of a child PLC

Accessing the *Set config* Screens

The first **Set config** screen is accessed as follows:

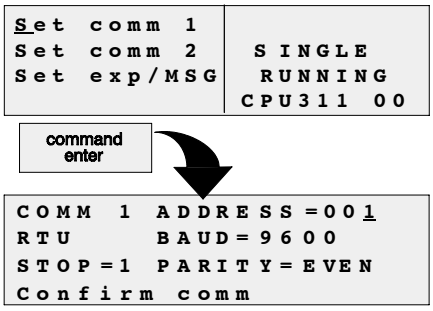


The cursor can be moved only through the menu selections on the left side of the screen.

Note If the PLC is a CPU311 or CPU411 model, the second menu selection, **Set comm 2**, does not appear on the screen. These models have only one RS-232 communications port.

Editing Comm Parameters on an RS-232 Port

If you place the cursor under either **Set comm 1** or **Set comm 2** and then push command enter, a new screen appears displaying the current port parameters assigned to the selected RS-232 port:



The process of editing RS-232 port parameters is the same regardless of the operating mode of the PLC.

You can edit the communication parameters for the selected port in the following order:

- The Modbus address of the RS-232 port(s): The Modbus address is an integer value in the range **1** ... **247**
- The communication protocol used by the port (7-bit **ASCII** or 8-bit **RTU**)
- The baud used on the RS-232 port(s): The baud may be set to **50**, **75**, **110**, **134**, **150**, **300**, **600**, **1200**,

1800, **2000**, **2400**, **3600**, **4800**, **7200**, **9600**, or **19200**

- The number of stop bits used the **comm2** port may be set to **1** or **2**; the stop bit must be set to **1** on **comm1**
- The parity check scheme on **comm2** may be set to **EVEN**, **ODD**, or **NONE**; the parity scheme on **comm 1** must be **EVEN**

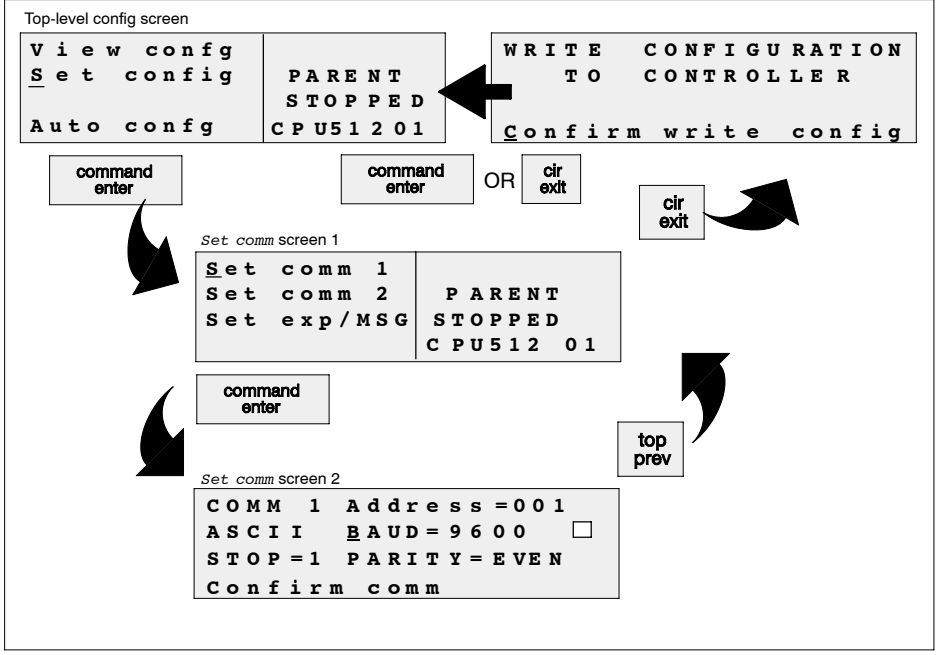
☞ **Note** If the comm port you are parameterizing is to be used with the HHP, the only valid bauds are **2400** when the **ASCII** protocol is used and **9600** when the **RTU** protocol is used, the only valid stop bit setting is **1**, and the only valid parity check scheme is **EVEN**. The other parameter choices are available so that the port may be set for use with other devices.

Whenever you change any of the comm parameters to settings that are not legal for the HHP, a small open square appears on the right side of the second line. For example, if you set the baud to **7200**:

```
COMM 1 ADDRESS=001
RTU    BAUD=7200   □
STOP=1 PARITY=EVEN
Confirm comm
```

Below is an illustration showing how to navigate through the screen sequence.

Setting the Communication Parameters for a *comm 1* Port



Here is a seven-step procedure for specifying communication parameters for the selected RS-232 port in this screen:

Step 1. To change the Modbus address,

move the cursor into the **Address** field and enter the desired value above the cursor—the current number will be overwritten. To confirm the value, push command enter ; to cancel the value and return to the previous one, push ‡ clr exit .

Step 2. Push foff to move the cursor to the communication mode field—the leftmost field on the second line of screen 2. You can toggle between **ASCII** and **RTU** with disa or enab .

Step 3. Push foff to move the cursor to the **BAUD** field on the same line. Use disa or enab to scroll to the desired baud value.

Step 4. Push foff to move the cursor to the **STOP** bit field on the third line of screen 2. Use disa or enab to scroll to the desired stop bit value.

Step 5. Push foff to move the cursor under the **PARITY** field on the same line. Use disa or enab to toggle to the desired parity value.

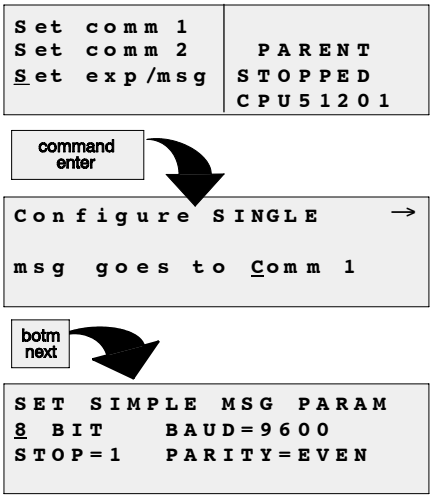
Step 6. When all the communication parameters for this port are set to the desired values, push clr exit .

Result You return to the first **Set config** screen where you may either edit other comm port parameters or exit out of the editor by proceeding to step 7.

Step 7. Push clr exit again. A screen appears prompting you to write the new configuration parameters to the PLC. To confirm the new parameters, push command enter ; to cancel the new parameters and leave the previous parameters intact, push clr exit again.

Setting Messaging Parameters

When you select **Set exp/MSG** from the **Set config** screen, a series of two screens appears. For example:



The first screen shows the current operating mode of the PLC and the port on the PLC that is set for simple messaging. Notice the arrow in the upper right corner of this screen, telling you that another screen follows in the series.

☞ **Note** The operating mode of the PLC—single, parent, or child—is determined by the previous configuration, and cannot be changed in a **Set config** operation.

Only one port on the PLC can be used for messaging. You can set the messaging port assignment to any one of the following:

- **Comm 1**
- **Comm 2**
- **Explink**
- **None**

To change the messaging port, scroll through the port choices with disa ◄ or enab ► .

Then push botm next . The next screen shows the comm port parameters to be used on the selected PLC port for simple messaging. The HHP allows you to specify a wide range of comm parameters for the simple messaging port:

Available Simple Messaging Port Parameters		
Comm Protocol	7 BIT (ASCII) 8 BIT (ASCII)	
Baud	50 ☐ * 75 ☐ * 110 ☐ * 134 ☐ * 150 ☐ * 300 ☐ * 600 ☐ * 1200	1800 ☐ * 2000 ☐ * 2400 3600 ☐ * 4800 7200 ☐ * 9600 19200 ☐ *
Stop bits	1 2	
Parity Check	EVEN ODD NONE	

* The open square appears on the screen next to baud values that cannot be supported when the HHP is used as the display panel in a simple messaging operation. When the HHP is the display panel, only bauds of 1200, 2400, 4800, and 9600 can be used.

The look of the **Set exp/MSG** screens differs depending on the PLC's operating mode.

Two examples follow—the first example shows an autoconfigured single PLC whose messaging port will be changed; the second example on page 40 shows an autoconfigured child whose I.D. number will be changed.

Example 1: Changing the Messaging Port and Comm Parameter on a PLC

In this first example, we look at a 110CPU51201 PLC in single mode, with its messaging capability on the **exp link** port and its communications protocol 8-bit ASCII. Using the following seven-step procedure, the messaging route is changed to the PLC's **comm 1** port and the protocol to 7-bit ASCII:

Step 1. Starting at the first config screen, place the cursor under **Set config** and push command enter.

Result. The following screen appears:

```
Set comm 1
Set comm 2          SINGLE
Set exp/MSG         STOPPED
                    CPU51201
```

The operating mode status information on the right side of the screen indicates that the PLC is in *single* mode.

Step 2. Place the cursor under **Set exp/MSG** and push command enter.

Result. The following screen appears:

```
Configure SINGLE                →
msg goes to Explnk
```

The information shows the default *single* mode parameter for messaging—i.e., the PLC port set for messaging is the **exp link** port.

Step 3. Place the cursor under **Explnk** and push enab.

Result. This keystroke scrolls the messaging port selection to **Comm 1**:

```
Configure SINGLE                →
msg goes to Comm 1
```

Step 4. Push botm next.

Result. This following screen appears:

```
SET SIMPLE MSG PARAM
8 BIT     BAUD=9600
STOP=1    PARITY=EVEN
```

Step 5. With the cursor under **8 BIT**, push enab to toggle to **7 BIT**. Then push cir exit.

Result. This previous screen appears:

```
Set comm 1
Set comm 2          SINGLE
Set exp/MSG         STOPPED
                    CPU51201
```

Step 6. Push cir exit again.

Result A screen appears prompting you to write the new messaging route to the PLC:

```
WRITE CONFIGURATION
   TO CONTROLLER

Confirm write config
```

Step 7. To write the new messaging route and port parameter to the PLC, push command enter.

Example 2: Setting Up a Child-mode PLC

In this second example, we look at a 110CPU51200 PLC previously configured as child #4. The messaging route is set up so that messages generated in the parent PLC will be sent to the child's **comm 2** port.

In the following eight-step procedure, we change its I.D. number to child #2 and change the messaging route so that the child will use it **comm 2** messaging port instead of the parent:

Step 1. Starting at the top-level config screen, place the cursor under **Set config** and push command enter .

Result. The following screen appears:

```
Set comm 1
Set comm 2     CHILD  4
Set exp/MSG    STOPPED
               CPU51200
```

The status information on the right side of the screen indicates that the PLC is child #4.

Step 2. Place the cursor under **Set exp/MSG** and push command enter

Result. The following screen appears:

```
Configure CHILD    →
CHILD no = 4
PARENT msg to Comm 2
```

Step 3. With the cursor under 4, push INI 2 , then command enter .

Result. The child I.D. number shown on the screen changes to 2:

```
Configure CHILD    →
CHILD no = 2
PARENT msg to Comm 2
```

Step 4. Push foff ▼ .

Result. The cursor moves under **PARENT** on the third line of the screen.

Step 5. Push enab → .

Result The selection toggles **CHILD**, indicating that messaging on the **comm 2** port will now be implemented by the child.

```
Configure CHILD    →
CHILD no = 2
CHILD msg to Comm 2
```

Step 6. Push cir exit .

Result. The original screen re-appears. Note that the status still shows the child I.D. as 4:

```
Set comm 1
Set comm 2     CHILD  4
Set exp/MSG    STOPPED
               CPU51200
```

Step 7. Push cir exit again.

Result A screen appears prompting you to write the new messaging route to the PLC:

```
WRITE CONFIGURATION
   TO CONTROLLER

Confirm write config
```

Step 8. To write the new child I.D number and messaging route to the PLC, push command enter .

Viewing Configuration Parameters

If you place the cursor under **View confg** in the first config screen and then push command enter, a set of status screens is displayed. These screens show the current parameters in the PLC configuration table.

☞ **Note** The **View confg** screens are all passive—i.e., you cannot edit any of the data in these screens.

Viewing an Unconfigured PLC

If the PLC is unconfigured when you select **View confg**, a screen appears with the following message:

```
View confg
Set config
Write cnfg
Auto confg first
```

```
SINGLE
STOPPED
```

command enter

```
NO CONFIGURATION
PRESENT
```

To return to the previous screen, push clr exit.

Viewing a Configured PLC

If the PLC is configured when you select **View confg**, four configuration status screens become available.

***View confg* Screen 1**

Here's an example of **View confg** screen 1:

```
SINGLE                              →
MSG GOES TO COMM 1
2400 7 BIT 1 STOP EV
AVAIL= 805 USED=1280
```

This first status screen tells you:

- The operating mode of the PLC—single, parent, or child
- The PLC port on which messaging communications will occur
- The communication parameters of the messaging port
- How much user program memory has already been used and how much is still available

The message on the second line, which describes the PLC port that handles the system's messaging capability, can vary depending on the PLC type and its operating mode.

☞ **Note** Remember that no more than one port on the PLC can be used to support the messaging capability.

The possible situations are shown in the following table:

Messages on Line 2 of *View Config* Screen 1		
Message	**Meaning**	**Mode/Model Restrictions**
`MSG GOES TO COMM 1`	The **comm 1** RS-232 port is being used for messaging	None
`MSG GOES TO COMM 2`	The **comm 2** RS-232 port is being used for messaging	PLC must be a 110CPU512 or 110CPU612 model
`MSG TO EXP LINK`	The RS-485 port is being used for messaging	PLC must be in single operating mode
`CHILD MSG TO COMM1`	The child PLC is using its own **comm 1** port for messaging	PLC must be in child operating mode
`CHILD MSG TO COMM2`	The child PLC using its own **comm 2** port for messaging	PLC must be in child operating mode and must be a 110CPU512 or 110CPU612 model
`PARENT MSG TO COMM1`	The parent is accessing **comm 1** on the child PLC and is using it for messaging	PLC must be in child operating mode
`PARENT MSG TO COMM2`	The parent is accessing **comm 2** on the child PLC and is using it for messaging	PLC must be in child operating mode and must be a 110CPU512 or 110CPU612 model
`MSG GOES TO NONE`	Messaging is not being used on any of the PLC's comm ports	None

If messaging is activated on one of the PLC ports, the third line of **`View confg`** screen 1 gives the communication parameters of the messaging port. Messaging always uses the **`ASCII`** communications protocol. Any one of four bauds is valid—**`1200`**, **`2400`**, **`4800`**, or **`9600`**. One stop bit, and an even parity check must be used. If messaging is *deactivated*, then the third line is empty.

The fourth line on **`View confg`** screen 1 lists the amount of user program memory available and the amount already used. The sum of the two equals the total amount of user program memory in your PLC.

Push cir exit if you want to return to the previous screen. Push botm next if you want to move forward to **`View confg`** screen 2.

View confg Screen 2

Here's an example of **`View confg`** screen 2:

```
   COMM 1          COMM 2 →
ADDRESS=01      ADDRES=02
2400  ASCII     9600   RTU
1  STP  EVEN    1  STP  EVE
```

This status screen shows the communication parameters of the **comm** RS-232 port(s) on your PLC. Those parameters include:

- The Modbus address of the RS-232 port(s): The Modbus address is an integer value in the range **`1`** ... **`247`**. Each address must be unique with respect to all other Modbus addresses on the same network.
- The baud and communication protocol used on the RS-232 port(s): Two protocols are valid—**`RTU`** and **`ASCII`**. The baud on **comm 1** must be **`9600`** for RTU and **`2400`** for ASCII. Any one of four bauds is valid for **comm 2** with either protocol—**`1200`**, **`2400`**, **`4800`**, and **`9600`**.
- The number of stop bits and the parity check scheme used on communications to or from the RS-232 port(s): **`1`** stop bit and **`EVEN`** parity must be used on **comm 1**. **`1`** or **`2`** stop bits

and **EVEN**, **ODD**, or **NONE** parity may be used on **comm 2**.

☞ **Note** For 311 or 411 models of the Micro, which do not have a **comm 2** port, the right side of the display is empty.

Push cir exit if you want to return to the first config screen. Push top prev if you want to return to **View confg** screen 1. Push botm next if you want to move forward to **View confg** screen 3.

View confg Screen 3

Here's an example of **View confg** screen 3:

```
0X  -  1536  1X  -  512 →
3X  -    48  4X  - 1872
2 SEGMENTS
MAX LOGIC  = 2085
```

This status screen shows you the PLC resources available to you for ladder logic programming:

- The number of discrete and register references available: Discrete outputs or coils are **0X** references, discrete outputs are **1X** references, register inputs are **3X** references, and register outputs are **4X** references.
- The number of segments available: This number is generally **2**, but may be any integer in the range **1** ... **32**.
- The total amount of user program memory: This number is the sum of the **USED** and **AVAIL** numbers shown in **View Config** screen 1.

☞ **Note** For 311 and 411 models of the Micro, the total amount of user program memory is about 1K words (a little more than 1K if you are not using a maximum configuration and I/O map, less than 1K if you have installed loadable instructions).

Push cir exit if you want to return to the first config screen. Push top prev if you want to return to **View confg** screen 2. Push botm next if you want to move forward to **View confg** screen 4.

View confg Screen 4

Here's an example of **View confg** screen 4:

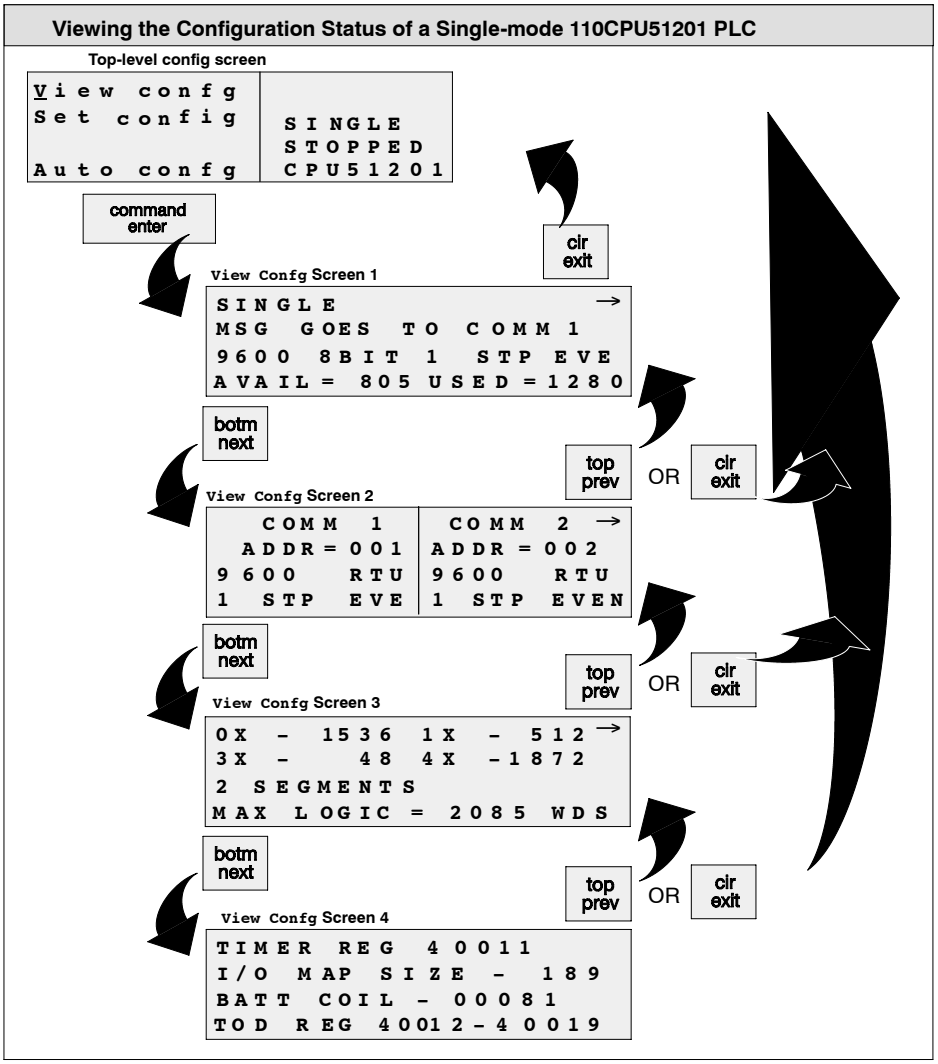
```
TIMER REG  40011
I/O MAP SIZE  -  189
BATT COIL - 00081
TOD REG 40012-40019
```

This status screen shows you:

- The reference for the holding register reserved for the free-running timer (the default is **40011**)
- The size—i.e., the number of words—in the PLC's I/O map
- The reference for the battery coil (the default is **00081**)
- The references for the eight holding registers where the time-of-day clock information is maintained (the defaults are **40012** ... **40019**)

Push cir exit if you want to return to the first config screen. Push top prev if you want to return to **View confg** screen 3.

Here is an example of the **`View confg`** screen/command sequence:



Chapter 4
Addressing I/O Locations with the I/O Map Editor

- Invoking the I/O Map Editor
- Default I/O Map Parameters
- Editing and Saving I/O Addresses
- Addressing a PLC's Fixed I/O Locations
- Sharing a Child's Fixed I/O Resources with the Parent
- Addressing Generalized Data Transfer Registers
- Addressing A120 I/O Expansion Modules
- Monitoring the Communications/Health Status of the PLC

Invoking the I/O Map Editor

The primary purpose of the I/O map editor is to let you associate the input and output signals used in the application to addresses in the PLC's memory. These addresses are reference 0*x*, 1*x*, 3*x* and 4*x* numbers.

Secondary functions of the I/O map provide you with health status information (always a read-only function) and allow you to edit the PLC's holdup time.

Four conditions must be true to support I/O map editing:

- The PLC must be stopped
- The HHP keyswitch must be positioned in *program* mode—if you are locked in *monitor* mode and the key is not available, contact your systems administrator for assistance
- If password security is enabled, you must have level 3 security access—contact your systems administrator if you need more information about security access
- The HHP must be *logged in* to the PLC—i.e., an HHP that is logged in can execute all HHP functions with respect to the PLC

☞ **Note** Two HHPs may be connected to the comm ports on a Micro PLC, but only one of the two can be logged in at any one time.

If any of these conditions is not true, the I/O map editor becomes read-only—you can view the current parameters in the I/O map but you cannot change them.

Accessing the I/O Map

The I/O map editor can be invoked from the top-level menu of the HHP if the PLC has a valid configuration. The **Map I/O** selection is on the second top-level menu screen:

Operations	Rev 1.00
Map I/O	SINGLE
Xfer/Save	STOPPED
Slave Hhp	CPU31100

Use the [toff ▼] key to move onto this selection, then push [command enter] .

You are now in the I/O map editor. The menu choices on the next screen depend on the configuration of the PLC.

Single and Child I/O Map Choices

If the PLC you are using has been configured as a single or child unit, a screen similar to this appears upon entry into the I/O map editor:

Self	Map I/O
	SINGLE
Comms/hlth	STOPPED
	CPU51201

The entries on the right of the screen are read-only status information. You cannot move the cursor into the right side of this screen.

The two entries on the left of the screen are menu picks:

- Selecting **Self** indicates that you want to monitor or edit the fixed or A120 I/O locations associated with the PLC with which the HHP is communicating; fixed resources include the various discrete and analog I/O points, high-speed counter/timer/interrupt inputs, and generalized data transfer capability built into the Micro PLCs
- Selecting **Comms/hlth** indicates that you want to display information that monitors and logs the health and communication status of the PLC

Parent I/O Map Choices

If the PLC you are using is configured as a parent unit, a third menu selection item—**Child** ***n*****/4**—appears on the left of the screen (where ***n*** may be **1**, **2**, **3**, or **4**):

Self	**Map I/O**
Child: 1/4	**PARENT**
Comms/hlth	**STOPPED**
	CPU51201

- Selecting **Child** and then specifying the number **1** ... **4** indicates that you want to monitor or edit the fixed I/O locations of the specified child PLC that are under the control of the parent

Menu Selection Process

Any one of these menu picks can be selected by moving the cursor under it with the fon ▲ and foff ▼ keys, then pushing command enter .

I/O Map Editor Conventions

When you select **Self** or **Child 1** ... **4**, the HHP displays the first in a series of I/O location (LOC) screens. A LOC may represent

- A fixed I/O resource of the PLC your HHP is connected to
- A fixed I/O resource of a child PLC being accessed by the parent
- An A120 I/O module driven by the PLC over the parallel I/O expansion link

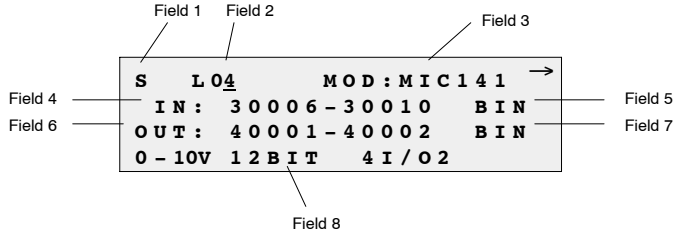
Each LOC is presented on an individual screen in the I/O map editor. The first five LOCs are reserved for the PLC's fixed resources as follows:

- LOC01 is reserved for the PLC's fixed discrete I/O points
- LOC02 is reserved for the fixed high-speed counter/interrupt inputs available on all model 411, 512, and 612 PLCs; if you are addressing I/O on a 311 model PLC, LOC02 should not be used
- LOC03 is reserved for the fixed counter/timer input register available on all 411, 512, and 612 model PLCs; if you are addressing I/O on a 311 model PLC, LOC03 should not be used
- LOC04 is reserved for fixed analog I/O channels available in the 612 model PLCs; if you are addressing I/O on a 311, 411, or 512 model PLC, LOC04 should not be used
- LOC05 is reserved for registers used in a generalized data transfer operation between parent and child PLCs; the child as addressed through the parent's I/O map should be given the same settings as the child addressed through its own I/O map

On the 512 and 612 model PLCs, an additional 15 locations (LOC06 ... LOC20) are provided to support A120 I/O modules driven by the PLC over the parallel I/O expansion link.

LOC Screen Layout

LOC screens can display data in as many as eight fields:



The top line of the LOC screen contains three data fields. Field 1 on the top left shows a letter or alphanumeric that indicates whether the I/O map you are viewing is for the PLC to which you are connected or for a child addressed through the parent:

- **S** indicates *self*—it appears whenever you are viewing the I/O map editor of a single or child PLC or the editor of a parent that addresses its own fixed and expanded I/O resources
- **C1** ... **C4** indicate child #1 ... child #4 when the child's fixed I/O resources are being accessed through the parent; the alphanumeric can only appear when you are using a parent PLC

Field 2 in the center shows the I/O location number **L *xx***. This number identifies the LOC*nn* resource number being addressed in this screen.

Field 3 on the right shows an **MIC** number (for fixed I/O) or A120 I/O model number (for expanded I/O) that indicates the type of I/O used in that location.

The second and third lines of the screen can each contain two additional data fields—field 4 on the left side of the second line lists the range of references to which the inputs are addressed, and field 5 on the right side of the second line lists the format in which the bits can be displayed (if the inputs are stored in a 3*x* register). If the references in field 4 are discretes, field 5 is blank.

Field 6 on the left side of the third line lists the range of references to which the outputs are addressed, and field 7 on the right side of the third line lists the format in which the bits can be displayed (if the outputs are stored in a 4*x* register). If the references in field 6 are discretes, field 7 is blank.

Field 8 on the fourth line of the LOC screen contains descriptive information regarding the **MIC** number selected in field 2.

Field 1 is set in the configuration process and cannot be edited in the I/O map editor. The cursor cannot be placed in field 1 in any LOC screen.

Fields 2 and 3 can be edited in any LOC screen. Fields 4 ... 7 may or may not be edited, depending on the nature of the I/O type. For example, if the LOC is only able to address high-speed inputs, then only field 4 can be edited—field 5 is unavailable if the inputs are mapped to discretes, and fields 6 and 7 are unavailable because the LOC does not address outputs.

In LOCs where both inputs and outputs can be addressed, fields 4 and 6 can be edited. In LOCs where the inputs are addressed to 3*x* registers, field 5 can be edited; in LOCs where the outputs are addressed to 4*x* registers, field 6 can be edited.

Field 8 is a read-only field. It changes automatically based on edits made in fields 2 or 3, but the cursor cannot be placed there and it cannot be edited directly.

Moving between Fields in a LOC Screen

You can move the cursor between fields with the fon ▲ and foff ▼ keys:

- To move forward through the fields, use foff ▼
- To move backward through the fields, use fon ▲

Both keys will wrap to the first or last operational field.

In the following example screen, the cursor is initially located in field 2 under **LOC02**:

```
S   L  04     MOD:MIC141 →
 IN:  30006-30010  BIN
OUT:  40001-40002  BIN
0-10V  12BIT   4 I/O 2
```

If you push foff ▼, the cursor move to field 3, under **MIC141**. Pushing foff ▼ a second time moves the cursor down to field 4 under **30006-30010**. Pushing foff ▼ a third time causes the cursor move down to field 5 and so on until the cursor reaches field 7, **BIN**, on the third line. The information in field 8 on the fourth line is status data, and you cannot move the cursor into it. If you push foff ▼ while the cursor is in field 7, the cursor wraps back to field 2 on the top line.

Moving from One LOC Screen to Another

With the cursor positioned in field 1, you can change the LOC screen in either of two ways:

- Type in a new location number, then push command enter
- Push botm next or top prev

In the example screen above, the cursor is located under **LOC02**. If you want to move forward to **LOC03**, simply push botm next. Alternatively, if you want to move backward to **LOC01**, push top prev.

If the cursor is positioned in **LOC01** when you push top prev, the HHP wraps to **LOC20**. Likewise, if the cursor is positioned at **LOC20** when you push botm next, the HHP wraps to **LOC01**.

If you want to move to a noncontiguous location in either direction, say **LOC12**, you can type in the number **12** above the cursor, then push command enter.

If you type a number larger than the number of LOCs available in the PLC configuration, say **25**, and push command enter, the HHP displays the original LOC number on the top line and the following message on the bottom line:

```
S   L  04   MOD:MIC140  →
 IN:  10081-10088
OUT:
E-locns 1-20 only
```

Note Whenever the HHP allows you to move forward to another LOC, an arrow appears in the upper right of the screen. When you have reached the last available LOC for you PLC—LOC05 for CPU311 and CPU411 models, LOC20 for CPU512 and CPU612 models—the arrow disappears.

Default I/O Map Parameters

When you specify the operating mode of the PLC—*single*, *parent*, or *child*—in the configuration process, the PLC is automatically assigned a set of default I/O map parameters. You may use these parameters to run your application or modify them in the I/O map editor to suit your application requirements.

The default parameter assignment is determined by the type of PLC and its operating mode.

Default Parameters for Single and Parent PLCs

If you have configured the PLC as a parent or a single, the default I/O map addresses the I/O locations as follows.

LOC01 Default Parameters

LOC01 maps all the fixed discrete inputs to sequential references starting at 10001 and all the fixed discrete outputs to sequential references starting at 00001.

Here are the default I/O type identifiers and discrete I/O resources for the various Micro PLCs:

PLC Type	I/O Type Identifier	Discrete I/O Resources
110CPU31100 110CPU41100 110CPU51200 110CPU61200	**MIC128**	16 24VDC in / 12 relay out
110CPU31101 110CPU41101 110CPU51201	**MIC131**	16 115VAC in / 8 triac out, 4 relay out
110CPU31102 110CPU41102 110CPU51202	**MIC134**	16 230VAC in / 8 triac out, 4 relay out
110CPU31103 110CPU41103 110CPU51203 110CPU61203	**MIC137**	16 24VDC in / 12 FET out

Here is a sample LOC01 screen:

```
S   L 01    MOD:MIC131   →
 IN:  10001-10016
OUT:  00001-00016
16@115V I/O 8TR/4RY
```

The **S** in field 1 on the top left of the screen indicates that the LOC01 I/O being addressed is for the PLC's *self*—i.e., its own discrete I/O points and not those of a child on the I/O expansion link. The default I/O type identifier, **MIC131**, is in field 3 on the top right of the screen. The reference addresses are shown in fields 4 and 6 on the second and third lines respectively, and the description of the fixed I/O resources is shown in field 8 on the bottom line of the screen.

Note In this example, references 00013 ... 00016 are expended in the addressing process. These references cannot be used as mapped to external coils or discrete outputs. They are, however, available for internal program use.

LOC02 Default Parameters

LOC02 on 411, 512, and 612 model PLCs is **MIC140**:

```
S   L 02    MOD:MIC140   →
 IN:  10081-10088
OUT:
8 INTPT/CNTR INP
```

This I/O type identifier addresses the high-speed inputs to discrete references 10081 ... 10088.

On 311 model PLCs, LOC02 should always **MOD:UNCNFG**:

```
S   L 02    MOD:UNCNFG   →
 IN:
OUT:
```

The cursor does not move beyond field 3—no references should be addressed there.

LOC03 Default Parameters

LOC03 on 411, 512, and 612 model PLCs is **MIC147**:

```
S   L  03    MOD:MIC147   →
 IN:  30001-30001   BIN
OUT:
16BIT TMR/CNTR VAL
```

This I/O type identifier addresses the counter/timer values to register 30001. The values can be displayed by default in binary format.

On the 311 model PLCs, LOC03 should always **MOD:UNCNFG**:

```
S   L  03    MOD:UNCNFG   →
 IN:
OUT:
```

The cursor does not move beyond field 3—no references should be addressed there.

LOC04 Default Parameters

LOC04 on a 612 model PLC is **MIC141**:

```
S   L  04    MOD:MIC141   →
 IN:  30006-30010   BIN
OUT:  40001-40002   BIN
0-10V 12BIT   4I/O2
```

This I/O type identifier defines the four analog inputs at 0 ... 10 V with 12-bit resolution. Input values from 0 ... 4095 correspond to 0 ... 10 V and voltages greater than 10 V result in over-range on the channel.

The analog channels are addressed as follows:

Analog Inputs

Input Register	Register Content
30006	Input channel 1
30007	Input channel 2
30008	Input channel 3
30009	Input channel 4
30010	status

Analog Outputs

Output Register	Register Content
40001	Output channel 1
40002	Output channel 2

On 311, 411, and 512 model PLCs, LOC04 should be **MOD:UNCNFG**:

```
S   L  04    MOD:UNCNFG   →
 IN:
OUT:
```

The cursor does not move beyond field 3—no references should be addressed there.

LOC05 Default Parameters

LOC05 in all Micro PLCs defaults to **MOD:UNCNFG**. because the use of generalized data transfer is an option between parent and child PLCs. When LOC05 is set up for GDT, it must be edited into the I/O maps of the child and of the parent. LOC05 must be set in the appropriate ***Cn*** screen of the parent's I/O map, where ***n*** is the number of the child PLC. (See pages 69 ... 71 for details.)

Default Parameters for A120 I/O LOCs

The model 512 and 612 PLCs are equipped with a parallel port that supports A120 I/O modules. These modules are addressed in LOC06 ... LOC20 of the PLC's I/O map.

If you have installed the A120 modules in their appropriate slots and powered them up, the LOC06 ... LOC20 screens in the I/O map editor will display the associated module identifier in field 3 on the top right of the screen. You then need to enter the desired input and/or output reference numbers on the second and/or third lines of the screen.

If you have not installed the A120 I/O modules or have not powered them up at the time that you are writing your I/O map, the LOC06 ... LOC20 screens come up showing **MOD:UNCNFG** in field 3.

If you are using a 311 or 411 model PLC, A120 I/O cannot be addressed. The I/O map editor does not allow you to access any LOCs beyond LOC05.

Default Parameters for a Child PLC

If you have configured the PLC as a child, the default I/O map shows **MOD:UNCNFG** in all locations. If you want all the fixed I/O resources of the child to be controlled by the parent, you do not need to edit the first five locations in the child's I/O map. Instead, these parameters can be edited in the parent's I/O map.

If you want to access some or all of the fixed I/O resources in the child you are viewing, you must edit the child I/O map to indicate the resources that will be accessed there. We will discuss the process of splitting I/O between the parent and child later in this chapter.

If you want to drive A120 I/O with any of the child PLCs in your system, you must edit the A120 information in the child I/O map—i.e., resources associated with a child PLC that cannot be accessed by the parent over the serial I/O expansion link.

Default Parameters for a Child PLC Addressed through its Parent

If you are addressing a child PLC's fixed I/O resources through its parent, the default parameters displayed by the HHP will vary depending on whether the child PLC is running or stopped.

If the child PLC is stopped while you are editing the I/O map, the default locations all show **MOD:UNCNFG**. You must edit the I/O map to indicate the fixed I/O resources of the child PLCs to be accessed by the parent.

If the child PLC is running while you are editing the I/O map, the default locations display useful information about the resources available to the parent (see pages 65 ... 68 for more detail).

Editing and Saving I/O Addresses

If you are using a PLC configured in parent or single operating mode, the default parameters of LOC01 ... 04 are usually good enough to use without any further editing. If you want to use the generalized data transfer capability of your PLC, you must set up the desired transfer register in fixed LOC05 of the I/O map editor; if not, simply leave LOC05 in its default **UNCNFG** condition.

If the PLC is configured in child operating mode and you want to place some or all of its fixed I/O resources under local control, you must edit the I/O map screens to suit your application requirements. By default, the child I/O map leaves all its fixed I/O initially unconfigured and accessible to the parent. You will need to edit both the parent and child if the two PLCs split any of the child's fixed I/O resources.

Accessing the LOC Screens

To edit the fixed I/O locations of any Micro PLC—single, parent, or child—select the **Self** menu selection on the first I/O map editor screen, then push

.

Editing References In an I/O Map Screen—An Overview

The reference numbers—which address the inputs and outputs in your physical system to memory locations in the PLC—appear in fields 4 and 6 on the second and third lines, respectively, of an I/O map screen. Input references are shown in field 4, and output references in field 6.

The HHP allows you to address physical inputs to either discrete 1*x* references or 3*x* input registers and physical outputs to either discrete 0*x* (coil) references or 4*x* output (holding) registers.

Mapping I/O to Discrete References

Discretes are mapped to contiguous references in blocks of 8 or 16. For example, if your PLC has 16 fixed discrete inputs and 12 fixed discrete outputs, the I/O map editor would show the inputs in LOC01 addressed to references 10001 ... 10016 and the outputs to 00001 ... 00016. Notice that although only 12 real outputs are used, 16 0*x* references are consumed in the mapping.

If, on the other hand, you have split your I/O so that only four fixed outputs are being used by the PLC you are addressing, the I/O map editor would show the outputs addressed to references 00001 ... 00008. In this case, you still have discrete references 00009 ... 00016 available for other addressing requirements.

Discrete references are addressed in sequences of multiples of 16 + 1:

10001–10016
10017–10033
10049–10064 ...

or multiples of 8 + 1:

00001–00008
00009–00016
00017–00024
00025–00032 ...

If you try to address discrete I/O to a block of references that is out of sequence—e.g., **00018–00034**—an error appears on the screen:

```
E-ref 00018, range
```

Out-of-range reference numbers are determined by the way your PLC is configured. If you are using the HHP default configuration values to address discrete I/O to references in a 311 or 411 model

PLC, the highest range of reference numbers you can specify is:

- **10241–10256** for discrete inputs
- **01009–01024** for discrete outputs

If you are using the HHP default configuration values to address discrete I/O to references in a 512 or 612 model PLC, the highest range of discrete reference numbers you can specify is:

- **10497–10512** for discrete inputs
- **01521–01536** for discrete outputs

If you try to address discrete I/O to a range of reference numbers that exceeds the maximum—e.g., **10513–10529**—an error appears on the screen:

```
E-ref 10513, range
```

Mapping I/O to Registers

Register references (3*x* registers for inputs, 4*x* registers for outputs) always contain 16 bits. The bits may be used to store representational data—e.g., count accumulation, timing—or to store groups of discrete I/O points so that they can be moved or manipulated as blocks in the logic program.

Whenever field 4 in an I/O map LOC screen addresses that LOC's inputs to a 3*x* register, field 5 beside it displays the kind of format the bits can be displayed in.

Whenever field 5 in an I/O map LOC screen addresses that LOC's outputs to a 4*x* register, field 6 beside it displays the bit conversion format. The bit format may be either a binary (**BIN**) or binary coded decimal (**BCD**) format. By placing the cursor in field 5 or field 6 of a register-mapped location, you can toggle between **BIN** and **BCD** formats with the disa ◀ or enab ▶ key.

Saving Changes in the I/O Map Editor

After you have made changes to one or more LOC screens in the I/O map editor, you can save the changes by pushing cir exit. A screen similar to this appears:

```
Save SINGLE   Map I/O
Changes for: Self
Holdup=0000300 msec
 Confirm or <exit>
```

The top two lines of this screen display status information about the operating mode of the PLC you are I/O mapping and whether you have addressed I/O locations for your own PLC or for a child on the expansion link.

The third line shows the current holdup time in hundreds of milliseconds. The default holdup time if no other has been explicitly specified is 300 ms. This number may be edited in this screen.

The fourth line prompts you to either confirm or ignore the changes you have made in the I/O map editor (and to the holdup time on the line above). By default, the cursor appears under the **Confirm** selection on the bottom line. To ignore the changes you have made, push cir exit.

Modifying the Holdup Time

When a parent and child PLC operate together on an I/O expansion link, the child constantly monitors the link to make sure that communications are ongoing. If the child senses that communications have failed, it waits for a certain period of time before it declares a comm error and sets all its outputs to 0. This period is the *holdup time*.

Generally, users like to keep the holdup time to a short duration—around

300 ms. However, there may be occasions where you want to increase the holdup time—for example, if you want the child to hold onto its current I/O values while you change a cable. The holdup time value can be changed in the save screen of the I/O map editor.

To change the holdup time, move the cursor to the third line of the screen by pushing fon ▲ or foff ▼. The cursor appears under the third digit from the right in the holdup time number:

```
Save SINGLE    Map I/O
Changes for: Self
Holdup=0000300 msec
  Confirm or <exit>
```

The default holdup time is 300 ms. Type in the number you want—the field fills from right to left from the cursor position. Push command enter, then push foff ▼ to move the cursor to the **Confirm** selection on the bottom line.

The resolution of the holdup time is in hundreds of milliseconds—e.g., 1500 ms is a legal holdup time value, but 350 ms is illegal. The minimum holdup time you can specify is 300 ms, and the maximum is 6,553,500 ms.

Saving I/O Map Changes

To save all the changes you have made in the I/O map editor, place the cursor under **Confirm** and push command enter. The HHP then returns you to the initial I/O map editor screen:

Self	Map I/O SINGLE
Comms/hlth	STOPPED CPU51201

where you can choose continue to edit or monitor I/O map parameters or exit the editor. To continue inside the editor, select either **Self** or **Comms/hlth**. To exit the editor, push clr exit.

Ignoring I/O Map Changes

To clear the changes you have made in the I/O map editor, push clr exit. The HHP then returns you to the initial I/O map editor screen:

Self	Map I/O SINGLE
Comms/hlth	STOPPED CPU51201

where you can choose continue to edit or monitor I/O map parameters or exit the editor. To continue inside the editor, select either **Self**/**Child** or **Comms/hlth**.

To exit the editor, push clr exit a second time.

Addressing a PLC's Fixed I/O Locations

When you enter the I/O map editor by selecting **Self**, the first five LOC screens you see (LOC01 ... 05) refer to the fixed I/O locations of the PLC to which your HHP is connected. If you are using a parent PLC and you enter the I/O map editor by selecting one of the following:

- **Child 1/4**
- **Child 2/4**
- **Child 3/4**
- **Child 4/4**

the five screens you see refer to the fixed I/O location of the selected child that will be accessed and controlled by the parent.

Editing LOC01

In the case of a parent or single PLC, you might want to change that PLC's LOC01 screen for a couple of reasons:

- To change the range of 1*x* input references and/or the range of 0*x* output references
- To re-address the discrete inputs to a 3*x* register or the discrete outputs to a 4*x* register
- To change the current **MIC** number, which defines the type of I/O that reside in that LOC

In the case of a child PLC, if you want to allow the parent to have access to all the LOC01 I/O resources, leave LOC01 unconfigured. On the other hand, if you want to split some of the LOC01 I/O resources with the parent or place all those resources under the child's own control, you must edit the I/O map accordingly.

Editing a Range of Discrete References

There may be times in an application when you want to address the fixed discrete inputs in your PLC to references other than 10001 ... 10016 or the fixed discrete outputs to references other than 00001 ... 00016.

Here is a sample five-step procedure for changing the range of inputs and outputs in the LOC01 screen:

Step 1. Push [foff ▼] to move the cursor to field 4 on line 2 of the screen.

Result The cursor appears under the leftmost digit in the low number of the input reference range. Here is an example using a 110CPU51201 PLC:

```
S   L  01    MOD:MIC131   →
 IN:  10001 -10016
OUT:  00001-0 0 016
16@115V  I/O  STR/4RY
```

Step 2. Type in the reference number you want associated with this first fixed input. Remember that this number must be the first reference number in a valid sequence—e.g., **10017**, **10033**, **10049**.

Step 3. Push [command enter] .

Result The HHP fills in the high reference number in the range. For example, if you type **10033**, the range in field 4 reads:

```
S   L  01    MOD:MIC131   →
 IN:  10033-10048
OUT:  00001-0 0 016
16@115V  I/O  STR/4RY
```

Step 4. To change range of output references, push [foff ▼] to move the cursor to field 6, then repeat step 2, this time using a 0*x* reference—e.g. **00017**.

Step 5. Push [command enter].

Result The new range of discrete outputs appears:

```
S   L  01    MOD:MIC131  →
 IN:  10033-10048
OUT:  00017-00032
16@115V  I/O  STR/4RY
```

☞ **Note** Fields 5 and 7 on the second and third lines, respectively, are left empty when the references are discretes. No bit conversion is available.

Re-addressing Discretes to Registers

The HHP allows you to re-address your discrete inputs in field 4 to a 3*x* input register and your discrete outputs in field 6 to a 4*x* holding register. For example, if you want to I/O map the default outputs in 110CPU51201 PLC to register 40027, follow this procedure:

Step 1. Push [foff ▼] to move the cursor to field 6 on line 3 of the screen.

Result The cursor appears under the leftmost digit in the low number of the output reference range:

```
S   L  01    MOD:MIC131  →
 IN:  10001-10016
OUT:  00001-00016
16@115V  I/O  STR/4RY
```

Step 2. Type in the register reference number you want associated with these outputs—**40027**.

Step 3. Push [command enter].

Result The HHP fills in the output range as follows:

```
S   L  01    MOD:MIC131  →
 IN:  10001-10016
OUT:  40027-40027
16@115V  I/O  STR/4RY
```

Step 4. Push [fon ▲] to move up to the second line on the screen, and change the input reference to a 3*x* register, say **30027**—so that the inputs and outputs are compatible. Then push [command enter].

Result The HHP fills in the input range as follows:

```
S   L  01    MOD:MIC131  →
 IN:  30027-30027
OUT:  40027-40027
16@115V  I/O  STR/4RY
```

Step 5. Push [clr exit] or [botm next] to exit the screen.

☞ **Note** If the outputs are addressed to a register, then the inputs must also be addressed to a register. Otherwise, an error message will appear when you try to exit the screen. For example:

```
S   L  01    MOD:MIC131  →
 IN:  10001-10016
OUT:  40027-40027
16@115V  I/O  STR/4RY
```

[clr exit]

```
S   L  01    MOD:MIC131  →
 IN:  10001-10016
OUT:  40027-40027
E-Inp & Out Incompat
```

Changing an MIC Number

It is possible for an inappropriate **MIC** number to appear in field 3 of the LOC01 I/O map screen. For example, if you are using a 110CPU51201 PLC, which has 115 VAC inputs and triac and relay outputs, the appropriate **MIC** number would be **MIC131**, **MIC132**, or **MIC133**. However, the I/O map could have been edited (or downloaded) so that **MIC137** appears (defining fixed 24 VDC inputs and FET outputs in the PLC).

To edit the **MIC** number, move the cursor into field 3 on the top line of the screen and scroll through your list of choices with the disa ◄ or enab ► key. Notice that each time the **MIC** number changes in field 3, the information in field 8 on the bottom line also changes to reflect the I/O description for that **MIC** number. The choices include all legal combinations of discrete I/O for all Micros:

PLC Type	Legal I/O Identifiers	Discrete I/O Resources
110CPU31100 110CPU41100 110CPU51200 110CPU61200	**MIC128**	16 24VDC in / 12 relay out
	MIC129	16 24VDC in / 8 relay out
	MIC130	16 24VDC in / 4 relay out
110CPU31101 110CPU41101 110CPU51201	**MIC131**	16 115VAC in / 8 triac out, 4 relay out
	MIC132	16 115VAC in / 8 triac out
	MIC133	16 115VAC in / 4 relay out
110CPU31102 110CPU41102 110CPU51202	**MIC134**	16 230VAC in / 8 triac out, 4 relay out
	MIC135	16 230VAC in / 8 triac out
	MIC136	16 230VAC in / 4 relay out
110CPU31103 110CPU41103 110CPU51203 110CPU61203	**MIC137**	16 24VDC in / 12 FET out
	MIC138	16 24VDC in / 8 FET out
	MIC139	16 24VDC in / 4 FET out

You also have the option of scrolling to a selection called **UNCNFG**, which puts the fixed discrete I/O points in an unaddressable state.

When you begin to scroll through the **MIC** choices, you will notice two things:

- The input and output reference ranges in fields 4 and 6 become empty
- A black square appears to the left of the **MIC** number—this black square indicates that you have scrolled to a selection that is available for this LOC but is not currently programmed

```
S   L  01 ■ MOD:MIC131   →
 IN:
OUT:
16@115V  I/O  8TR/4RY
```

To change the **MIC** number in LOC01, follow this three-step procedure:

Step 1. Cursor to the desired **MIC** number—it will have a black square to its left because it is currently unprogrammed.

Step 2. Change in the I/O ranges in fields 4 and 6 of the screen, using the previous procedure.

Step 3. Push command enter .

At this point, you may either go to another LOC screen in the I/O map editor or initiate an exit from the editor by pushing clr exit . When you exit the I/O map editor, you will be asked to confirm or ignore all the changes you have made while you were in the editor.

Editing LOC02

If you are using a single or parent PLC that supports the fixed high-speed interrupt inputs associated with LOC02, these inputs are addressed by default to references 10081 ... 10088:

```
S   L  02   MOD:MIC140  →
 IN:  10081-10088
OUT:
8  INTPT/CNTR  INP
```

There are four kinds of changes that can be made at LOC02:

- Unconfiguring the high-speed inputs
- Changing the range of the discrete 1*x* references
- Addressing the inputs to a 3*x* input register instead of eight discrete 1*x* references
- Changing the data format of the register by toggling the entry in field 5

Unconfiguring the High-speed Inputs

In all single and parent Micro PLCs that support fixed high-speed inputs, the default I/O type identifier for LOC02 is **MIC140**. If you do not want to address the inputs in the I/O map, you can unconfigure them via the following procedure:

Step 1. Push [fon ▲] or [foff ▼] to move the cursor into field 3 on the top line under **MIC140**.

Step 2. Push [disa ◄] or [enab ►] to scroll to the optional choice **UNCNFG**.

Result The data in fields 4 ... 8 are cleared as shown here:

```
S   L  02   MOD:UNCNFG  →
 IN:
OUT:
```

and the cursor cannot be moved into any of these fields—i.e., pushing [disa ◄] or [enab ►] only toggles between fields 2 and 3.

Notice also that the black square appears to the left of field 3, indicating that the current selection is available but has not been saved.

Step 3. Push [clr exit], then follow the instructions for saving the I/O map changes.

Changing the Range of Discrete References

There may be times in an application where you want to use the high-speed inputs but you do not want them addressed to references 10081 ... 10088. Here is a sample four-step procedure that changes the discrete input range to 10025 ... 10032:

Step 1. Make sure that field 3 is set to **MIC140**, then push [foff ▼] to move the cursor to field 4.

Result The cursor appears under the leftmost digit in the low number of the input reference range:

```
S   L  02   MOD:MIC140  →
 IN:  10081-10088
OUT:
8  INTPT/CNTR  INP
```

Step 2. Type in the reference number you want associated with this first fixed input. The number must be a multiple of 8 + 1.

Step 3. Push [command enter].

Result The HHP fills in the high reference number in the range. For example, if you type **10025**, then command enter, the range in field 4 reads:

```
S   L  02    MOD:MIC140   →
 IN:  10025-10032
OUT:
8  INTPT/CNTR  INP
```

Step 4. Push clr exit, then follow the instructions for saving the I/O map changes.

Addressing the Inputs to a 3*x* Register

The sample six-step procedure that follows re-addresses the discrete inputs in LOC02 to a 3*x* register and changes the bit format for the register's content to BCD:

Step 1. Make sure that field 3 is set to **MIC140**, then push foff ▼ to move the cursor to field 4.

Result The cursor appears under the leftmost digit in the low number of the input reference range:

```
S   L  02    MOD:MIC140   →
 IN:  10081-10088
OUT:
8  INTPT/CNTR  INP
```

Step 2. Type in the 3*x* reference number you want to store the eight high-speed inputs in—e.g., **30010**.

Step 3. Push command enter.

Result Since a single register is all that is required to store eight bits, the screen looks like this:

```
S   L  02    MOD:MIC140   →
 IN:  30010-30010   BIN
OUT:
8  INTPT/CNTR  INP
```

Step 4. To change the bit format in register 30010, push foff ▼.

Result The cursor move to field 5 under the bit format indicator (the default is **BIN**):

```
S   L  02    MOD:MIC140   →
 IN:  30010-30010   BIN
OUT:
8  INTPT/CNTR  INP
```

Step 5. Push clr exit to toggle the format from **BIN** to **BCD**.

Step 6. Push clr exit, then follow the instructions for saving the I/O map changes.

Editing LOC03

If you are using a single or parent PLC that supports the fixed counter/timer inputs associated with LOC03, then by default the values are accumulated in register 30001:

```
S   L  03    MOD:MIC147   →
 IN:  30001-30001   BIN
OUT:
16BIT  TMR/CNTR  VAL
```

Similarly to the LOC02 screen, the LOC03 screen can be edited in four ways:

- Unconfiguring the counter/timer register for your application
- Changing the 3*x* input reference
- Changing the data format of the register by toggling the entry in field 5

Re-addressing the counter/timer inputs to a series of 16 contiguous discrete 1*x* references

Unconfiguring the Counter/timer Register

If you do not want to address the counter/timer values in the I/O map, you can unconfigure them via the following procedure:

Step 1. Push fon ▲ or foff ▼ to move the cursor into field 3 on the top line under **MIC147**.

Step 2. Push disa ◄ or enab ► to scroll to the optional choice **UNCNFG**.

Result The data in fields 4 ... 8 are cleared as shown here:

```
S   L  03  MOD:UNCNFG  →
 IN:
OUT:
```

and the cursor cannot be moved into any of these fields —i.e., pushing disa ◄ or enab ► only toggles between fields 2 and 3.

Notice also that the black square appears to the left of field 3, indicating that the current selection is available but has not been saved.

Step 3. Push cir exit , then follow the instructions for saving the I/O map changes.

Changing the Register Reference

There may be times in an application where you want to use the counter/timer values but you need to store them in an input register other than 30001.

Here is a four-step procedure for changing the reference number in the LOC02 screen:

Step 1. Make sure that field 3 is set to **MIC147**, then push foff ▼ to move the cursor to field 4.

Result a screen like this appear:

```
S   L  03   MOD:MIC147  →
 IN:  30001-30001   BIN
OUT:
16BIT  TMR/CNTR  VAL
```

Step 2. Type in the reference number you want associated with this first fixed input. The number can be any 3*x* register in the range 30001 ... 30048.

Step 3. Push command enter .

Result The HHP fills in field 4 as shown below; for example, if you type **30045**, then push command enter :

```
S   L  03   MOD:MIC147  →
 IN:  30045-30045   BIN
OUT:
16BIT  TMR/CNTR  VAL
```

Step 4. To change the bit format in register 30045, push foff ▼ .

Result The cursor move to field 5 under the bit format indicator (the default is **BIN**):

```
S   L  03   MOD:MIC147  →
 IN:  30045-30045   BIN
OUT:
16BIT  TMR/CNTR  VAL
```

Step 5. Push cir exit to toggle to the desired format—the two choices are **BIN** and **BCD**.

Step 6. Push [clr exit], then follow the instructions for saving the I/O map changes.

Addressing the LOC03 Values to Discrete References

The sample four-step sample procedure that follows re-addresses the LOC03 values from register 30001 to discrete references 10049 ... 10064:

Step 1. Make sure that field 3 is set to **MIC147**, then push [toff ▼] to move the cursor to field 4.

Step 2. Type in the reference number you want associated with this first fixed input—in this case, **10049**.. This number must always be a multiple of 8 + 1—i.e., 10009, 10017, 10025, etc.

Step 3. Push [command enter].

Result The HHP fills in the high reference number in the range:

```
S   L  02    MOD:MIC147   →
 IN:  10049-10064
OUT:
16BIT TMR/CNTR VAL
```

Step 4. Push [clr exit], then follow the instructions for saving the I/O map changes.

Editing LOC04

LOC04 is reserved for addressing the fixed analog I/O resources offered only on the 612 model PLCs. This location should be left unconfigured if you are using any other model PLC (or if you are mapping a child PLC whose analog I/O will be totally accessed by the parent).

If you are addressing fixed I/O in LOC04, the four input channels are addressed by default to register 30006 ... 30010 and the two output channels to registers 40001 and 40002. The default I/O type identifier, **MIC141**, defines the range of the input channels as 0 ... 10 V with 12-bit resolution in both the input and output channels:

```
S   L  04    MOD:MIC141   →
 IN:  30006-30010   BIN
OUT:  40001-40002   BIN
0-10V 12BIT   4I/O2
```

This screen can be edited in several ways:

- Change the **MIC** number and thereby change the voltage range and resolution of the analog input channels
- Change the four 3*x* registers to which the analog input channels are addressed
- Change the two 4*x* registers to which the analog output channels are addressed
- Change the data format of the input registers by toggling the entry in field 5 and/or the output registers by toggling the entry in field 7

Editing MIC Numbers and Register Ranges

To edit the **MIC** number, move the cursor into field 3 on the top line of the screen and scroll through your list of choices with the [disa ◄] or [enab ►] key. Notice that each time the **MIC** number changes in field 3, the information in field 8 on the bottom line also changes to reflect the input voltage range and bit resolution associated with that **MIC** number.

The choices available as you scroll through this menu include all legal com-

binations of discrete I/O for all models of Micro PLC:

Analog In Identifiers	Input Voltage Range	Channel Resolution
MIC141	0 ... 10 V	12-bit in / 12-bit out
MIC142	1 ... 5 V	12-bit in / 12-bit out
MIC143	± 10 V	12-bit in / 12-bit out
MIC144	0 ... 10 V	15-bit in / 12-bit out
MIC145	1 ... 5 V	14-bit in / 12-bit out
MIC146	± 10 V	16-bit in / 12-bit out

You also have the option of scrolling to a selection called **UNCNFG**, which makes the analog channels unaddressable.

When you begin to scroll through the **MIC** choices, you will notice three things:

- The input and output reference ranges in fields 4 and 6 become empty
- The range and resolution descriptions in field 8 on the bottom line change to describe what is available with each **MIC** choice
- A black square appears to the left of the **MIC** number—this black square indicates that you have scrolled to a selection that is available for this LOC but is not currently programmed.

Here is a sample procedure to change the **MIC** number in LOC04 from **MIC141** to **MIC144** and re-address the I/O channels to 30006 ... 30010 and 40001 ... 40002, respectively.

Step 1. Push fon or foff to move the cursor into field 3 on the top line under **MIC141**.

```
S   L  04    MOD:MIC141   →
 IN:  30006-30010   BIN
OUT:  40001-40002   BIN
0-10V 12BIT   4I/O2
```

Step 2. Push disa or enab to scroll to the desired **MIC** number—e.g., **MIC144**:

```
S   L  01 ■ MOD:MIC144   →
 IN:          -         BIN
OUT:          -         BIN
1-5V    14BIT   4I/O2
```

Notice the black square to the left of field 3, indicating that **MIC144** is not currently residing in this LOC. Notice also that the register data in fields 4 ... 7 has been cleared, but that the range and resolution data in field 8 on the bottom line has changed to reflect the **MIC144** identity.

Step 2. Push foff to move the cursor to field 4 on the second line of the screen.

Step 3. Type the first 3*x* register reference in the desired range—e.g., **30006**—in field 4, then push command enter .

Result The HHP fills in field 4 as follows:

```
S   L  04    MOD:MIC144   →
 IN:  30006-30010   BIN
OUT:        -       BIN
1-5V   14BIT   4I/O2
```

Step 4. Push foff once more to move the cursor to field 6 on the third line of the screen.

Step 5. Type the first 4*x* register reference in the desired range—

e.g., **40001**—in field 6, then push command enter .

Result The HHP fills in field 6 as follows:

```
S   L  04    MOD:MIC144   →
 IN:  30006-30010   BIN
OUT:  40001-40002   BIN
1-5V   14BIT   4I/O2
```

Step 6. Push cir exit , then follow the instructions for saving the I/O map changes.

Changing Register Formats

The format in which the register bits in fields 4 and 6 can be displayed is signified by the entries in fields 5 and 7 respectively. By default, the format is binary (**BIN**). All the registers in the range (four 3*x* registers, two 4*x* registers) have the same format.

If you move the cursor into field 5 or field 7 with the toff ▼ key, you can toggle between **BIN** and **BCD** formats. with disa ◄ or enab ► .

Sharing a Child's Fixed I/O Resources with the Parent

The I/O map parameters for the fixed I/O resources of a child PLC (LOC01 ... LOC05) are left unconfigured in the default condition. When you are addressing the child's I/O in an expanded I/O link, there are three ways you can set them up:

- Leave the fixed I/O unconfigured in the child itself and address it all through the parent
- Split the child's fixed I/O resources between it and the parent
- Address all the child's fixed I/O resources in its own I/O map, then use generalized data transfer to send data to and receive data from the parent over the expansion link (more on GDT later in this chapter)

Addressing a Child's I/O in the Parent

The Micro PLCs allow you to run your ladder logic program exclusively in the parent when multiple PLCs are used together via the expanded I/O link The parent is able to use the fixed I/O resources of its child PLCs like drops of remote I/O.

To set up your system this way, leave the I/O map screens in the child PLCs defaulted to **UNCNFG** states and address all I/O points through the parent. To access the I/O map of a child through the parent, go to the first screen in the parent's I/O map editor, select the second menu option on the left of the screen, and type the number of the child you want to access.

For example, if your control system uses four child PLCs and you want to access child #2, push toff ▼ to move the cursor under the second menu selection, and type **2**:

```
Self          | Map I/O
Child: 2 /4   | PARENT
Comms/hlth    | STOPPED
              | CPU51201
```

Then push command enter . The LOC01 I/O map screen of child #2 appears with the indicator **C2** in field 1 in the upper left corner of the screen. This indicator is always seen as a **C*n*** alphanumeric when you are accessing a child through its parent's I/O map editor.

For example, if child #2 is a 110CPU31101 model and you are accessing it through its parent, the LOC01 screen would look like this initially:

```
C2 L 01   MOD:UNCNFG  →
 IN:
OUT:
```

The PLC model you are using has 16 fixed discrete 115 VAC inputs, eight fixed triac outputs, and four fixed relay outputs. The I/O type identifier that specifies this mix of discretes is **MIC131**.

To put all the fixed discrete resources of child #2 under the control of the parent, push toff ▼ to move the cursor to field 3 under **MOD:UNCNFG**, then use disa ◄ or enab ► to scroll to the **MIC131** selection in the list of **MIC** numbers. One of the following two screens appears:

```
C2 L 01 □MOD:MIC131 →
 IN:
OUT:
16@115V I/O STR/4RY
```

or

```
C2 L 01 ■MOD:MIC131 →
 IN:
OUT:
16@115V I/O STR/4RY
```

The difference between the two screens is the color of the box to the left of field 3 on the top line. An unfilled box (□) indicates that the HHP does not detect any I/O at LOC01 in child #2—the child may not yet be installed or it may not yet be powered up and running. A filled box (■) indicates that the HHP detects I/O available at LOC01 in child #2 but it has not been saved to the I/O map as a **MIC131** mix.

Push toff ▼ to move the cursor to field 4 on the second line of the screen,. Type the first discrete reference in the range of 1*x* references that will be the addresses for the 115 VAC inputs. For example, if you want to address the inputs to references 10017 ... 10032, type **10017**. Then push command enter :

```
C2 L 01 ■MOD:MIC131 →
 IN: 10017-10032
OUT: _
16@115V I/O STR/4RY
```

The cursor moves to field 6 on the third line of the screen. Here you type in the first of the discrete output references and again push command enter . If you type **00017**, the screen looks like this:

```
C2 L 01 ■MOD:MIC131 →
 IN: 10017-10032
OUT: 00017-00032
16@115V I/O STR/4RY
```

To save the new I/O map parameters, push either:

- cir exit to exit the LOC screens when the new information is saved
- botm next / top prev to move to another LOC screen when the new information in this LOC screen is saved

When the parameters are saved, all the fixed discrete I/O resources of child # 2 will be placed under the control of the parent PLC.

You can re-enter the I/O map editor, again by selecting the **child 2/4** menu pick, and view the newly saved parameters for child #2. If you were previously editing a screen with a filled box (■), there should now be no box next to field 3, since the new parameters are now known by the parent PLC:

```
C2 L 01  MOD:MIC131 →
 IN: 10017-10032
OUT: 00017-00032
16@115V I/O STR/4RY
```

If you were previously editing a screen with an unfilled box (□), the unfilled box should still be there after you save the parameters. The parent PLC still does not recognize the presence of any I/O at the location because the child is not powered up:

```
C2 L 01 □MOD:MIC131 →
 IN: 10017-10032
OUT: 00017-00032
16@115V I/O STR/4RY
```

Splitting a Child's Fixed I/O Resources

The Micro PLCs also allow you to split the fixed I/O resources of child PLCs with the parent over the expanded I/O link. In this way, ladder logic programs can be run in both the parent and the child PLCs.

The following table shows the **MIC** numbers that can be used in the I/O map editors of the parent and child to split the I/O properly:

A child with the fixed resources:	Can split its resources to:	With these MIC numbers:
16 24VDC in / 12 relay out (**MIC128**)	16 24VDC in / 8 relay out	**MIC129**
	16 24VDC in / 4 relay out	**MIC130**
16 115VAC in / 8 triac out & 4 relay out (**MIC131**)	16 115VAC in / 8 triac out	**MIC132**
	16 115VAC in / 4 relay out	**MIC133**
16 230VAC in / 8 triac out & 4 relay out (**MIC134**)	16 230VAC in / 8 triac out	**MIC135**
	16 230VAC in / 4 relay out	**MIC136**
16 24VDC in / 12 FET out (**MIC137**)	16 24VDC in / 8 FET out	**MIC138**
	16 24VDC in / 4 FET out	**MIC139**

As you can see from the table, the inputs can actually be shared by the parent and child; the outputs must be split.

An Example

Let's suppose that child #1 on the expanded I/O link is a 110CPU31102 with eight fixed triac outputs and four fixed relay outputs. We want edit the I/O maps of the two PLCs to put the relays under the control of the parent PLC and we want to keep the triacs under the control of child #1.

Let's begin by connecting the HHP to the child and accessing LOC01 in the I/O map editor by selecting **Self** on the first I/O map screen:

```
Self          | Map I/O
              |  CHILD 1
Comms/hlth    | STOPPED
              | CPU31102
```

command enter

```
S   L 01    MOD:UNCNFG  →
 IN:
OUT:
```

Note The **s** (self) status information in field 1 on the top left of the screen indicates that you have accessed child #1 through its own I/O map editor.

Push toff ▼ to move the cursor to field 3 under **MOD:UNCNFG**, then use disa ◄ or enab ► to scroll to the **MIC135** selection. **MIC135** defines the I/O mix as 16 230 VAC inputs and eight triac outputs:

```
S   L 01  MOD:MIC135  →
 IN:
OUT:
16@230V I/O  8TR
```

Push toff ▼ to move the cursor into field 4 on the second line. Type in the first 1*x* reference number, say **10001**, and push command enter :

```
S   L 01  MOD:MIC135  →
 IN:  10001-10016
OUT:
16@230V I/O  8TR
```

Then push [toff ▼] to move the cursor into field 6 on the third line, type the first 0*x* reference number, and push [command enter] again:

```
S   L  01  ■MOD:MIC135   →
 IN:  10001-10016
OUT:  00001-00008
16@230V  I/O   8TR
```

Now push [clr exit] and save the new I/O map parameters. When these parameters are saved, the 16 fixed discrete inputs and the eight fixed triac outputs of child #1 will be placed under the child's control.

Now start the child PLC, disconnect the HHP from the child, and connect it to the parent. In this example, the parent is a 110CPU51203 PLC Access the I/O map editor of child #1 through the parent by selecting **Child 1/4** on the first I/O map screen:

```
Self              | Map I/O
Child: 1/4        | PARENT
Comms/hlth        | STOPPED
                  | CPU51203
```

[command enter]

```
C1  L  01    MOD:UNCNFG   →
 IN:
OUT:
```

☞ **Note** The **C1** status information in field 1 on the top left of the screen indicates that you have accessed child #1 through the parent's I/O map editor.

Push [toff ▼] to move the cursor to field 3 under **MOD:UNCNFG**, then use [disa ◄] or [enab ►] to scroll to the **MIC136** selection. **MIC136** defines the I/O mix as 16 230 VAC inputs and four relay outputs:

```
C1  L  01  ■MOD:MIC136   →
 IN:
OUT:
16@230V  I/O   4RY
```

Push [toff ▼] to move the cursor into field 4 on the second line. Type in the first 1*x* reference number, say **10001**, and push [command enter]:

```
C1  L  01  ■MOD:MIC135   →
 IN:  10001-10016
OUT:
16@230V  I/O   4RY
```

The cursor moves into field 6 on the third line, type the first 0*x* reference number, and push [command enter] again. Because the output range requires only four references, you can use multiples of 8 + 1—for this example, type **00009**;

```
C1  L  01  ■MOD:MIC135   →
 IN:  10001-10016
OUT:  00009-00016
16@230V  I/O   4RY
```

Now push [clr exit] and save the new I/O map parameters. When these parameters are saved, the full complement of discrete I/O points will be split between child #1 and its parent. Both PLCs will share the 16 inputs, child #1 will control the eight triac outputs, and the parent will control the four relay outputs.

Addressing Generalized Data Transfer Registers in a Parent and Child

Generalized data transfer (GDT) functionality requires that you do register setup in LOC05 of the child's I/O map editors accessed through itself and through the parent PLC. GDT should not be implemented if the PLC is in single mode.

To set up a GDT in the parent, you must access the I/O map LOC screens via the **Child: 1/4** menu selection. To do GDT setup in a child, you must access the I/O map of the same child specified in the parent I/O map.

In both LOC05 screens, the number of I/O words reserved for GDT must be the same. The HHP allows you to set up the following numbers of GDT words via the **MIC** number selection:

GDT Word Identifiers	Number of Words of I/O
MIC148	1 in / 1 out
MIC149	2 in / 2 out
MIC150	4 in / 4 out
MIC151	8 in / 8 out

A word may be a 3*x* input register, a 4*x* output register, a group of 16 contiguous 1*x* discrete references, or a group of 16 contiguous 0*x* discrete output references.

A GDT Example

Let's say that we want to implement four input registers, 30021 ... 30024, in child #3 to receive generalized data from holding registers 40025 ... 40028 in its parent PLC. Likewise, we want to set up input register 30011 ... 30014 in the parent to receive generalized data from holding registers 40015 ... 40018 in child #3.

Setting Up the Child for GDT

First, connect the HHP to child #3 and access its I/O map editor. Enter the editor by selecting **Self** on the first I/O map screen. Then advance to the LOC05 screen.

By default, this screen is initially unconfigured:

```
S   L  05    MOD:UNCNFG  →
 IN:
OUT:
```

Note The **S** (self) status information in field 1 on the top left of the screen indicates that you have accessed child #3 through its own I/O map editor.

In this example, we use four words of input and four words of output over the GDT connection. To define the number of words (registers) to be used, push toff ▼ to move the cursor to field 3 under **MOD:UNCNFG**, then use disa ◀ or enab ▶ to scroll to the **MIC150** selection:

```
S   L  05    MOD:MIC150  →
IN:
OUT:
4 WORD       I/O 4 WORD
```

The status line on the bottom of the screen tells you that you have selected four input word and four output words for GDT.

Push [toff ▼] to move the cursor to field 4 on the second line of the screen and type **30021**, followed by [command enter]. The HHP fills in the range of four contiguous 3*x* registers:

```
S   L  05    MOD:MIC150   →
IN:   30021-30024   BIN
OUT:
4 WORD    I/O 4 WORD
```

Push [toff ▼] again to move the cursor to field 6 on the third line of the screen and type **40011**, then [command enter]. The HHP fills in the range of four contiguous 4*x* registers:

```
S   L  05    MOD:MIC150   →
IN:   30021-30024   BIN
OUT:  40015-40018   BIN
4 WORD    I/O 4 WORD
```

The child PLC is now set up to handle GDT. Now connect the HHP to the parent.

Setting Up the Parent for GDT

With the HHP connected to the parent, access its I/O map editor. Enter the editor by selecting **Child 3/4** on the first I/O map screen:

```
Self              | Map I/O
Child: 3 /4       | PARENT
Comms/hlth        | STOPPED
                  | CPU51201
```

Then advance to the LOC05 screen. By default, this screen is initially unconfigured:

```
C3  L  05    MOD:UNCNFG   →
  IN:
OUT:
```

☞ **Note** The **C3** status information in field 1 on the top left of the screen indicates that you have accessed child #3 through the parent's I/O map editor.

We now need to set up the parent with four words of input and four words of output similar to child #3.

To define the number of words (registers) to be used, push [toff ▼] to move the cursor to field 3 under **MOD:UNCNFG**, then use [disa ◄] or [enab ►] to scroll to the **MIC150** selection:

```
C3  L  05    MOD:MIC150   →
IN:
OUT:
4 WORD    I/O 4 WORD
```

☞ **Note** The PLCs at both ends of the GDT connection must be set up with the same number of registers reserved for the transfers.

Push [toff ▼] to move the cursor to field 4 on the second line of the screen and type **30011**, followed by [command enter]. The HHP fills in the range of four contiguous 3*x* registers:

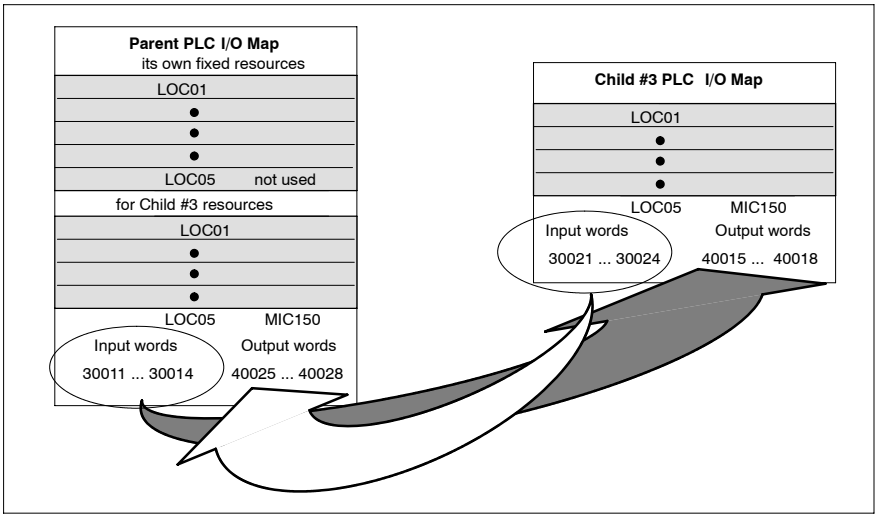
```
C3  L  05    MOD:MIC150   →
IN:   30011-30014   BIN
OUT:
4 WORD    I/O 4 WORD
```

Push [toff ▼] again to move the cursor to field 6 on the third line of the screen and type **40021**, then [command enter].

The HHP fills in the range of four contiguous 4*x* registers:

```
C3   L  05  MOD:MIC150   →
IN:   30011-30014   BIN
OUT:  40025-40028   BIN
4 WORD    I/O  4 WORD
```

The illustration below shows the generalized data transfer link you have set up between the parent and child #3 PLCs.



Addressing A120 I/O Expansion Modules

If you are using 512 or 612 models of the Micro PLC, you have the ability to address up to 15 additional A120 modules in your I/O map editor. Locations for A120 I/O are in LOC06 ... LOC20.

To be able to address A120 I/O, you must enter the I/O map editor via the **Self** menu selection. The HHP does not allow you to access locations beyond LOC05 when you access the I/O map of a child through its parent.

Accessing A120 I/O LOCs

Once you enter the I/O map editor by selecting **Self**, the HHP automatically displays the LOC01 screen. You can get to LOC06 and beyond quickly by simply typing the number of the LOC you want with the cursor placed in field 2, then pushing command enter .

If you enter a number larger than **20** in field 2, the HHP keeps the current LOC and displays an error message on the bottom line of the screen:

```
E-locs 1-20 only
```

To clear the message, type in another number.

How the I/O Map Editor Displays an A120 I/O LOC

The HHP displays A120 I/O data differently, depending on whether the module is physically present or not at the time you are editing the I/O map

Addressing an Installed A120 Module

If you are addressing an A120 module that is already installed and powered up in the LOC you have selected, the HHP will sense the identity of that module and display that information in the LOC screen.

For example, if you go to LOC06 in the I/O map and there is DEP210 input module installed and under power in the slot 1 of an A120 I/O rack, the screen would look like this:

```
S   L  06   MOD:DEP210   →
IN:
OUT:
115VAC  8-IN
```

You simply need to move the cursor to field 4 on the second line of the screen, type the first discrete input reference (1*x*) in a group of eight, and push command enter . For example, if you type **10033** and push command enter , the screen looks like this:

```
S   L  06   MOD:DEP210   →
IN:   10033-10040
OUT:
115VAC  8-IN
```

The HHP does not let you move the cursor to the third line because the DEP210 module is strictly an input module. Continue with other I/O modules in other LOCs.

When you are satisfied with the new parameters you have entered in the I/O map, push cir exit , then follow the instructions for saving the I/O map changes.

Addressing an Unpowered A120 Module

If you are addressing an A120 module that is either not yet installed or is in-

stalled but is not powered up, the HHP will not recognize the module and will display an **UNCNFG** LOC screen:

```
S   L  06   MOD:UNCNFG   →
IN:
OUT:
```

To specify the A120 I/O module that you want to use, move the cursor in to field 3 under **MOD:UNCNFG** and push disa ◄ or enab ► to scroll to the module you want to address in this LOC. The full range of A120 I/O modules is available to you.

For example, if you scroll to **DAP212** in LOC06, the screen looks like this:

```
S   L  06 □MOD:DAP212   →
IN:
OUT:
24VDC  4-OUT   8-IN
```

Note The unfilled square to the left of field 3 on the top line indicates that this module is unknown to the PLC. It will remain on the screen until the DAP212 module is installed in the proper location and powered up—even after the I/O map has been edited and the changes have been saved to the PLC.

The DAP212 module supports both inputs and outputs. Move the cursor to field 4 on the second line of the screen to enter your input range, then field 6 on the third line to enter your output range:

```
S   L  06 □MOD:DAP212   →
IN:   10017-10024
OUT:  00017-00024
24VDC  4-OUT   8-IN
```

When you are satisfied with the new parameters you have entered in the I/O map, push cir exit , then follow the instructions for saving the I/O map changes.

Checking the Number of Points Used for A120 I/O

The amount of I/O that can be addressed to A120 modules in a PLC is limited to 256 input points and 256 output points (or 16 registers in / 16 registers out). To assure proper operations, it is important that you not exceed these limits in your I/O map.

To check the amount of I/O resources currently addressed to A120 I/O, push the subr data key. a message appears on the bottom line of the screen telling you how many input points and how many output points have already been used from the 256 maximum. For example:

```
S   L  13   MOD:DAP212   →
IN:   10025-10033
OUT:  00025-00033
A120:in=144  out=128
```

the message on the bottom line tells you that your I/O map has used 144 input points and 128 output points for addressing A120 I/O. You still have the ability to address A120 I/O to another 128 output points and 112 input points with this PLC.

Monitoring the Communications/Health Status of the PLC

On the first screen of the I/O map editor is a menu option that sends you to a set of comprehensive screens where you can monitor the health and communication status of the PLC(s) in your system.

Health status is displayed as an array of symbols indicating whether or not the PLC you are connected to recognizes the presence of I/O in all its available locations. Communication status is displayed as a series of counters that update the frequency of comm errors in your system when it is running.

☞ **Note** The health status of I/O locations can be monitored in PLCs in any operating mode—single, parent, or child. The comm error counters, which monitor the relationship between parent and child PLCs on an expanded I/O link, can be accessed only through a PLC in parent mode.

Accessing the Comms/ Health Displays

To access the comms/health screens, select **Comms/hlth**, the last menu pick on the left side of the first I/O map editor screen:

```
Self          | Map I/O
Child: 1/4    | PARENT
Comms/hlth    | RUNNING
              | CPU51201
```

command enter

```
   Comms/Health
Place hlth    | PARENT
Err cntrs     | RUNNING
              |
```

The two entries on the left of the screen directly above allow you to access either the I/O location health status screens (**Place hlth**) or the error counter screens (**Err cntrs**).

Monitoring Health Status of the I/O Locations

You can monitor the health status of the parent, child, or single PLC to which your HHP is connected, and, if you are connected to a parent, you can monitor the health and status of all child resources being accessed by the parent.

To access these screens, push foff ▼ to move the cursor under **Place hlth**, then command enter :

```
   Comms/Health
Place hlth    | PARENT
Err cntrs     | RUNNING
              |
```

command enter

```
PARENT        | L01-05  111--  →
 I/O          | L06-10  -----
HEALTH        | L11-15  -----
              | L16-20  -----
```

The screen example directly above shows the health status of the fixed I/O locations (**LC01-05**) of a parent PLC and the status of the locations that can support A120 I/O modules (**LC06-20**). The number **1** indicates that the PLC recognizes healthy I/O at the given location, and the symbol **-** indicates that an unhealthy or unconfigured location has been detected.

A location will be shown as unhealthy if:

- The location is unconfigurable—e.g., LOC04, which addresses analog I/O channels, is unconfigurable on a 512 model PLC, and LOC05, which is reserved for generalized data transfer, cannot be used in the fixed I/O location of a parent or single PLC
- There is no I/O present in the given location—e.g., if A120 I/O is not being used in the application, LOC06 .. LOC20 have no modules present
- There is physical I/O present in the location but it has not been addressed in the PLC's I/O map
- There is physical I/O present in the location and it has been addressed in the PLC's I/O map, but it is not currently powered up
- There is physical I/O present in the location but it has been incorrectly identified in the PLC's I/O map—e.g., the first slot in the A120 I/O rack contains a powered-up DAP212 module but the module I/O addressed to LOC06 in the I/O map is a DAP220

A quick glance at the screen above shows that fixed locations LOC01 ... LOC03 are addressed in the PLC's I/O map and that A120 I/O is either not being used or has been installed but is not been powered up. If A120 I/O is installed and under power, then the PLC's I/O map and the physical modules in place do not match.

When the PLC you are monitoring is a parent, a second I/O health screen can be accessed by pushing botm next. This screen shows the health status of the fixed I/O resources of the child PLCs on the I/O expansion link as they are viewed from the perspective of the parent:

```
PARENT I/O HEALTH | L01-05 111-- →
                  | L06-10 -----
                  | L11-15 -----
                  | L16-20 -----
```

botm next

```
EXPNDS I/O HEALTH | Child1 1---1
                  | Child2 1----
                  | Child3 1----
```

In the example screen directly above, the parent has been configured with three child PLCs. The parent is accessing some or all of the fixed discrete I/O resources of all three child PLCs—i.e., LOC01 in all three is configured. Also child #1 has been set up to conduct generalized data transfer with the parent—i.e., LOC05 in child #1 is configured.

☞ **Note** Both the parent and child PLC must be running in order for the HHP to properly display I/O health. Also, the parent's I/O map must be correctly configured to match the child's I/O map when child I/O is shared.

To toggle back to the previous I/O health screen, push botm next again.

If the PLC you are monitoring is not in parent operating mode, the second I/O health screen cannot be accessed.

If the PLC you are monitoring is a 311 or 411 model, A120 I/O modules are not supported, and only the fixed I/O locations are displayed on the first I/O health screen:

```
PARENT I/O HEALTH | L01-05 111-- →
```

Monitoring the Comm Error Counters

You can monitor a series of screens, each showing six counters that record comm errors between the parent and each child on an expanded I/O link that is running. To access these screens, connect the HHP to the parent and access its I/O map editor.

Push [foff ▼] to move the cursor under **Err cntrs**, then [command enter]:

```
    Comms/Health
Place hlth     PARENT
Err cntrs      RUNNING
```

[command enter]

```
 COMM   W/CHILD 1      →
Retrys 023 Lost   005
No rsp 016 Prty   000
Framng 000 Ovrun  000
```

[botm next]

```
 COMM   W/CHILD 2      →
Retrys 000 Lost   000
No rsp 000 Prty   000
Framng 000 Ovrun  000
```

[botm next]

```
 COMM   W/CHILD 3      →
Retrys 000 Lost   000
No rsp 000 Prty   000
Framng 000 Ovrun  000
```

[botm next]

```
 COMM   W/CHILD 4
Retrys 000 Lost   000
No rsp 000 Prty   000
Framng 000 Ovrun  000
```

Notice that you can page through the various **COMM W/CHILD** screens with the [botm next] key. If you push [botm next] on the last screen, **COMM W/CHILD 4**, the HHP wraps back to the **COMM W/CHILD 1** screen.

Six three-position counters are shown in each screen. Two counters are shown per line on the second, third, and fourth lines of the screen. These counters log occurrences of the following communication errors between the parent and child:

- Communication retries (**retrys**)
- No responses within about the last holdup time period (**No rsp**)
- Framing errors (**Framng**)
- Lost communications (**Lost**)
- Parity errors (**Prty**)
- Communication overruns (**Ovrun**)

Each counter can of log up to 255 occurrences of the condition it is monitoring. While the PLCs are running, the counters dynamically accumulate the occurrences of the following conditions between the parent and the specified child.

You can access these screens while the parent PLC is in a stopped state, but the counter values will either be frozen with the values that were current when the PLC was stopped or all set to **000**.

☞ **Note** Under ideal conditions, all counters read **000**. Changing numbers in the counter indicate that either the child PLC is not running or there are problems in the parent-child communications.

To exit the comms/health screen, push [cir exit].

Chapter 5
Ladder Logic Programming

- Logic Edit Mode
- The Logic and Zoom Displays
- Entering Coils, Contacts, and Shorts with an HHP
- How the HHP Displays Contacts, Coils, and Shorts
- Entering Ladder Logic Instructions with an HHP
- How the HHP Displays Ladder Logic Instructions
- Errors of Range and Type
- Special Commands for Editing Logic
- Searching Nodes for Program Data

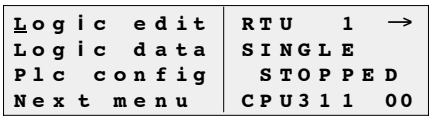
Logic Edit Mode

A ladder logic program can be monitored and edited with a Hand-held Programmer in logic edit mode. The HHP must be connected to a PLC that is powered up and configured before it can be put into logic edit mode.

Accessing Logic Edit Mode

Logic edit mode can be accessed from the main LCD screen or from a logic data screen (see Chapter 6).

To access logic edit mode from the main LCD screen, place the cursor under the first menu pick—**Logic edit**—on the left of that screen:

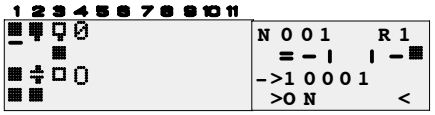


then push command enter .

To access the logic edit mode from a logic data screen, push the subr data key.

If the PLC Contains a Logic Program

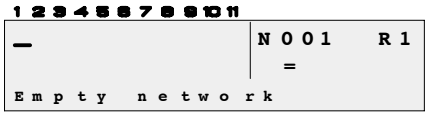
The LCD screen will display the top four rows of the first logic network in the program—e.g.:



where the symbols on the left represent the node content in the logic network, and the status information on the right describes the individual node at the cursor position in the network.

If the PLC Does Not Contain a Logic Program

The following screen appears in the LCD:



The cursor appears in the top left corner of the screen; its position indicates that your current location in a ladder logic network is in column 1. The status information on the top right of the screen tell you that the cursor is positioned in network 1 (**N001**) on row 1 (**R1**).

Cursor Movement in Logic Edit Mode

The cursor can be moved through the network display—the first 11 columns on the LCD screen—and can be used to select individual nodes on that screen.

The cursor will never appear in the zoom display on the right side of the LCD screen (columns 12 ... 20).

The cursor can be moved up, down, left, and right using the arrow keys, located in the top right corner of the HHP keypad.

You can move the cursor up and down a column in a network display using the fon and foff keys.

If you push fon when you are in row 1 or foff when you are in row 7, no cursor action will be effected—i.e.,

fon and foff do not wrap cursor around the column, and they do not move the cursor across network boundaries.

You can move the cursor left and right across a row in the network display with disa and enab .

If you push enab when you are in column 11, the cursor will wrap to column 1 in that row. If you push disa when you are in column 1, the cursor will wrap from to column 11 in that row.

Note The HHP allows you to move the cursor to the right of a coil in the same row where it is displayed, but it does not allow you to enter logic elements in the row to the right of the coil.

Moving within a Network

Remember that the network display can show only four rows of a seven-row network. You can push ‡ top prev and ‡ botm next to anchor the cursor at the top or bottom row of the network:

‡ top prev	Moves the cursor to row 1 of the network and positionss row 1 of the network in row 1 of the display
‡ botm next	Moves the cursor to row 7 of the network and positions row 7 of the network in row 4 of the display

Moving between Networks and Segments in Logic

You can move between networks in the logic program with the following keys:

botm next		Takes you to the next network in the program
top prev		Takes you to the previous network in the program
back get	*network number*	Takes you to the specified network in the program
‡ subr data		Takes you to the first network in the subroutine segment
‡ subr data	*with the cursor under a* ***JSR***	Takes you to the specified subroutine in the last segment

Paging Sequentially through the Networks

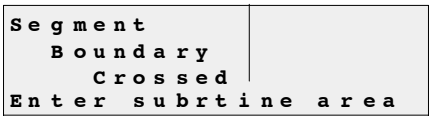
You can advance forward sequentially through the networks in a logic program by pushing botm next . To return backward sequentially through the networks in a logic program, push top prev . As you page backward or forward through the networks, the network number appears in the top line of the zoom display.

If you enter a network that does not contain any logic, the message **`Empty network`** appears on the bottom line of the screen.

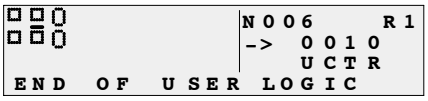
When you pass through a segment boundary, a screen like this appears:

```
Segment
   Boundary
      Crossed
Enter segment 02
```

where the message on the bottom line tells you which segment is being entered. If you are entering the last segment in logic—i.e., the subroutine segment—the following screen appears:



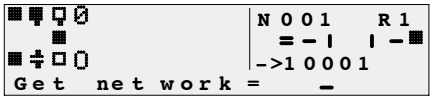
If you push botm next while you are in the last network in logic, the current screen display remains with the message **End of user logic** on the bottom line. For example:



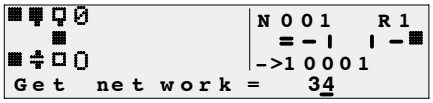
If you push top prev while you are in the first network in logic, the current screen display remains with the message **Strt of user logic** on the bottom line. For example:

Going Directly to the Desired Network

You can advance directly to a desired network in ladder logic from any other network by pushing the back get key. For example, if you are currently positioned in network 1 and you want to go to network 34, the following screen message appears when you push back get :

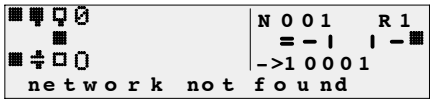


Simply type the desired network number—in this example **34**—at the cursor location on the bottom line:



and then push command enter . The HHP immediately displays network 34 in the logic display with the cursor in row 1 column 1.

If network 34 does not exist in the ladder logic program, the current screen remains displayed with an error message on the bottom line:



Creating a New Network

To create a new network positioned immediately after the one you are currently located in, push ‡ +net 0 no . If the current network is **N001**, the following screen will appear when you push ‡ +net 0 no :

If other networks previously followed **N001** before this new **N002** was created, they will now be positioned as one network further down in the segment—i.e., the former **N002** now becomes **N003**, the former **N003** becomes **N004**, etc.

These messages will be cleared by the next keystroke.

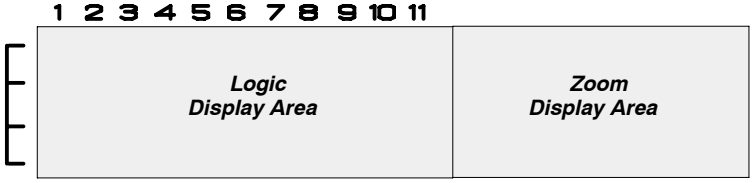
Toggling between Logic Edit and Logic Data Modes

Whenever you are in logic edit mode, you can switch to logic data mode by simply pushing subr data .

The Logic and Zoom Displays

Let's look closer at the way logic information is displayed on the LCD screens. When you are in logic edit mode, the screen is divided into two displays—a *logic* display on the left and a *zoom* display on the right. The two displays are separated by a vertical line drawn down through column 12.

The columns in the logic display are numbered 1 ... 11 above the top of the LCD—these correspond to the 11 columns in a ladder logic network. To the left of the LCD is a ladder logic power rail with indicators for four rows within the network. Because a network can contain as many as seven rows of logic, the LCD can give you only a partial view of the network's depth.

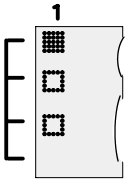


Node Symbols in the Logic Display

Symbols are used to indicate the presence of logic node elements in the logic display. The symbols change depending on whether or not the node is passing power. Fourteen symbols are used:

	Node without power		Vertical short without power
	Node with power flow		Vertical short with power flow
	Node with vertical short; no power flow through either		Horizontal and vertical shorts without power
	Node with vertical short; power flow through both		Horizontal and vertical shorts with power flow in both
	Node with vertical short; power flow through the short but not the node		Horizontal and vertical shorts with power flow in the vertical but not the horizontal
	Horizontal short without power		Coil without power
	Horizontal short with power flow		Coil with power flow

The symbols do not discriminate between single-node elements and individual nodes within two-high or three-high instructions. For instance, if column 1 of the logic display shows:



the three nodes could represent three contacts or three one-high instructions, one three-high instruction, or some combination of a contact or one-high instruction with a two-high instruction.

To view the content of any individual node, move the cursor under the symbol in the logic display and look in the zoom screen on the right side of the LCD.

Entering Coils, Contacts, and Shorts with an HHP

Coils, contacts, and shorts can be entered in a ladder logic network directly from explicit keys in the lower left corner of the HHP keypad. Notice that each entry is preceded by ‡ :

Key Sequence	Meaning	Appearance in Zoom Display	Can Replace
‡ ( ) C	A normal coil	—( )—	—(M)—
‡ (M) 3	A memory-retentive coil	—(M)—	—( )—
‡ I I A	A normally open (N.O.) contact	—| |—	—/— —| P |— —| N |—
‡ И B	A normally closed (N.C.) contact	—|/|—	—| |— —| P |— —| N |—
‡ IPI 1 yes	A positive transitional contact	—| P |—	—| |— —/— —| N |—
‡ INI 2	A negaitive transitional contact	—| N |—	—| |— —/— —| P |—
‡ – D	A horizontal short	———	A null node
‡ I F	A vertical short	|	A vertical open
‡ vopn E	A vertical open		A vertical short

Legal Entry Locations

A coil can be placed in an open node on any row and in any column in the logic display—no more than one coil can be placed in an individual row. The coil is the only logic element allowed in column 11 of a network.

Contacts and horizontal shorts can be placed in open nodes in any row and in any column except column 11. They cannot be placed in a row to the right of a coil.

Vertical shorts can be placed in open nodes on any row except row 7 and in any column except column 11. They cannot be placed to the right of a coil in the same row or in the row directly above the one in which the coil resides. They also cannot be placed in a column

directly above the node in which a coil resides.

It is also legal to overwrite one type of coil or contact with another type of coil or contact. For example, you can overwrite a normal coil with a memory-retentive coil or an N.O. contact with an N.C. contact. Column 4 of the table on the previous page lists the elements that can be replaced by the corresponding element type.

Entering a Logic Element in a Network

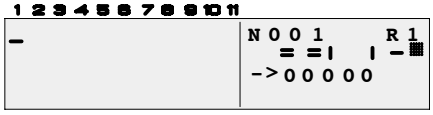
To enter a logic element, follow this four-step procedure:

Step 1. Move the cursor to a legal node location in the logic display area of the LCD with fon ▲ , foff ▼ , disa ◀ , and enab ▶ .

Step 2. Push the desired key sequence (see the table on the previous page). For example, if you want to enter an N.O. contact, push ‡ and A .

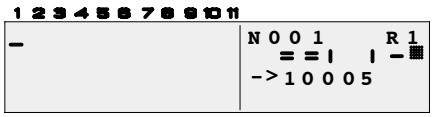
Result. As soon as you push ‡ , a ‡ symbol appears in column 12 of the top line of the zoom screen, indicating that the next character entered from the keypad will be the superior (red) character. Once that character is entered, ‡ disappears.

The symbol for the element you want to create appears on the second line of the zoom display, and an arrow on the third line of the zoom display points to a **00000** reference number. For example, if you enter an N.O. contact in the first node of row 1, network 1, the screen will look like this:



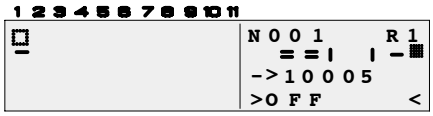
Step 3. Type the reference number you want associated with the new logic element.

Result. The reference number will appear after the arrow in line 3 of the zoom screen. If you type **10005** as the reference for an N.O. contact, for example, the screen looks like this:



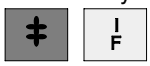
Step 4. Push command enter .

Result. A node symbol appears above the cursor in the logic display area, indicating that the element has been inserted in the network. The state of the new element appears in line 4 of the zoom display:



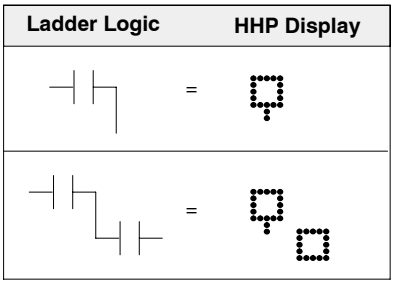
Entering a Vertical Short

The vertical short is a special case in terms of how it is entered in the logic screen. Unlike all other logic elements, the vertical short can share a node space with another element or instruction node. Therefore, it can be entered by moving the cursor back onto a node that already has logic in it and pushing



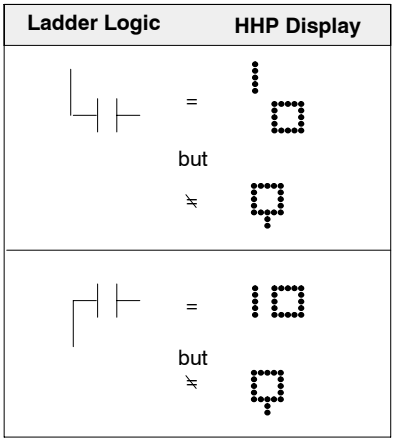
.

Power flow in a vertical short is always from the output of the element whose node it shares, never to the input. For example, contacts with associated vertical shorts would look something like this in the HHP display:



Of course, vertical shorts can also be entered by themselves in logic nodes similarly to other logic elements. Again, power flow is always downward through a vertical short.

The following illustration shows how the HHP will and will not display certain arrangements of contacts and vertical shorts:



How the HHP Displays Contacts, Coils, and Shorts

If you place the cursor under a contact, coil, or short that appears in the network display, the zoom screen will display:

- The number of the network displayed—i.e., your current location with respect to the first network in the controller
- The row number of the selected node—i.e., its location in the network—for example, if the cursor is under a node in the top row of the logic network display and the zoom screen tells you that it is in row 4, you know that you are viewing rows 4 ... 7 in the network
- The type of the selected element, its input, and its output (see the table below)
- The reference number of the selected node
- The ON/OFF state of the selected node
- In the case of a coil, whether it is enabled or disabled

The table below shows how a selected contact, coil, or short is displayed on the second line of the zoom screen. The display shows a combination of the element type along with its input and its output, where appropriate.

The input shows whether the element is receiving logic power flow. The output show when the element is passing power and the state of any vertical short that might follow it.

Symbols Displaying Contacts, Coils, and Shorts in the Zoom Screen		
Input	**Logic Element**	**Output**
(Power in) (No power in)	(Enabled normal coil with power) (Enabled normal coil without power) (Disabled normal coil) (Enabled memory-retentive coil with power) (Enabled memory-retentive coil without power) (Disabled memory-retentive coil)	N/A
	(Normally open contact) (Normally closed contact) (Positive transitional contact) (Negative transitional contact)	(Passing power) (Not passing power) (Passing power with a powered vertical short) (Not passing power with a powered vertical short) (Not passing power with an unpowered vertical short)
	(Horizontal short passing power)	(Passing power with no vertical short) (Passing power with a vertical short)
	(Horizontal short without power)	(Not passing power, no vertical short) (Not passing power with a powered vertical short) (Not passing power with an unpowered vertical short)
N/A	N/A	(Vertical short without power) (Vertical short passing power)

If, for instance, you select a normally open contact that is passing power and that outputs to a vertical short, the zoom screen would display that element as follows:

Certain combinations of the above inputs and outputs cannot occur together.

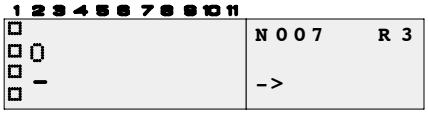
☞ **Note** Elements have vertical connectivity in the downward sense only.

Since power flow through the network is only allowed in the forward direction, an element cannot pass power out if it does not receive power from the input—e.g., a display such as

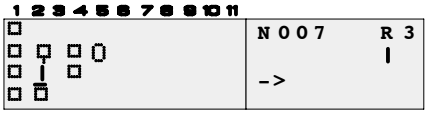
is impossible and will never be seen.

Displaying Null Nodes

When the cursor is moved onto an unpopulated node in the network display, the zoom display simply shows the node and row locations on its top line and an arrow on the third line pointing to no data:



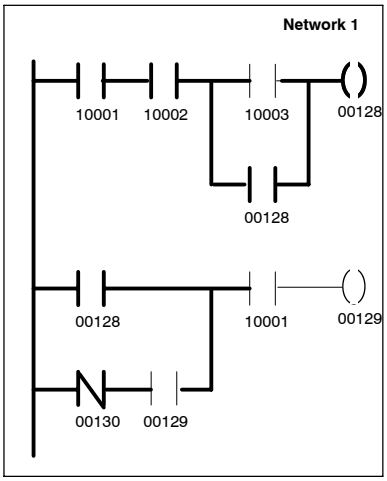
If the null node has vertical connectivity, that status information is shown on the second line of the zoom display:



An Example Network

Let's take a look at a ladder logic network that comprises contacts, coils, vertical shorts, and horizontal shorts. The network is four rows deep, so that its entire depth can be seen in a single network display, and it is four columns long.

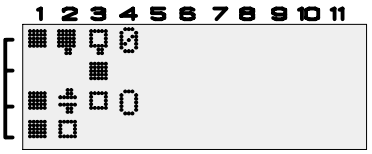
The network shown below shows how it looks when it is laid out on paper, with the network power flow indicated by darkened lines:



In the first and second rows of the network, power is passed through contacts 10001 and 10002 down the vertical to contact 00128 and back up to turn ON coil 00128.

In the third and fourth rows, power is passed through contacts 00128 and 00130. Contact 10001 receives power but does not pass it, so coil 00129 is turned OFF.

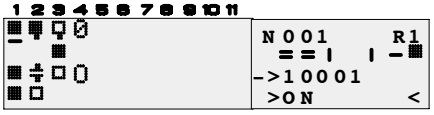
Here is how the network is shown in the network display of the HHP:



Power flow through the nodes is shown by the darkened node boxes and the double-lined shorts.

Notice that coils are always displayed in the column where they are solved on the HHP, not expanded to column 11 as they may be in some panel software displays.

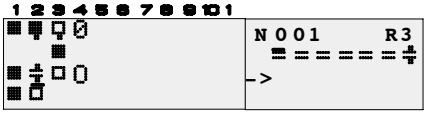
If we move the cursor under the top left node in the network, the LCD screen shows the following:



The top line of the zoom display gives you some navigational information about the location of the network and the selected node. It tells you that the node is in network #1 (**N001**) and in row 1 (**R1**) of that network. You therefore know that you are at the top of the network and at the beginning of the logic segment.

The next three lines in the zoom display give you information about the selected node. The second line shows that the node is a normally open contact passing power. the third line tells you that its data reference is 10001, and the bottom line tells you that it is ON.

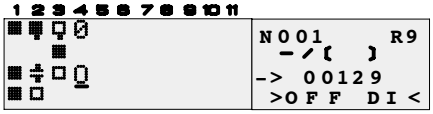
If you move the cursor to row 3, column 2, the node is displayed this way:



The zoom display tells you that you are now in row 3 (**R3**) of network 1 and that

you are looking at a horizontal short that is passing power to a vertical short. No further status information is necessary for this logic element.

When showing coils that have been disabled, the zoom display provides that information in two places. For example, if coil 00129 is disabled and the cursor is placed in row 3, column 4, the node is displayed this way:



The backslash symbol preceding the parentheses in the second line of the zoom display indicates that the coil is disabled. Also, the **OFF DI** status information on the fourth line tells you that coil 00129 is OFF and disabled.

Entering Ladder Logic Instructions with an HHP

The ladder logic instruction set can be accessed by scrolling through a series of six functional categories:

- Counter instructions
- Timer instructions
- Math instructions
- DX move instructions
- DX matrix instructions
- Other instructions.

Each functional category can be accessed by pushing one of the following key sequences. Notice that each entry is preceded by ‡ :

Key Sequence	Functional Category	Instructions
‡ cntr 4	counters	UCTR DCTR
‡ timr 5	timers	T1.0 T.01 T0.1 T1MS
‡ math 6	math	ADD DIV SUB EMTH MUL
‡ move 7	DX move	R→T FIN T→R FOUT T→T SRCH BLKM STAT
‡ mtrx 8	DX matrix	AND SENS OR MBIT XOR COMP CMPR BROT
‡ othr 9	others	SKP TBLK COMM CKSM PID2 BLKT LAB CTIF JSR SCIF RET

Legal Entry Locations

A single- or multi-node instruction can be placed in open nodes on any rows or in any column except column 11, as long as there are no coils to the left on any of the rows used by that instruction.

In most cases, the network in which the instruction is placed can be in any segment of ladder logic. Two exceptions are the **LAB** and **RET** instructions, which must be placed in the last segment (the subroutine segment) of ladder logic. **LAB** instructions must be placed in row 1, column 1 of the applicable networks in the subroutine segment.

A multi-node (two-node or three-node) instruction must be place in vertically contiguous nodes, and all the nodes of an individual instruction must fit in the same network. For example, the top node of a three-node instruction cannot be placed in row 6 of a network because it requires two additional rows below it and the network provides only one. Likewise, the top node of a three-node instruction cannot be placed in row 1 of a network column if there is a contact in row 2 or 3 of that column.

If you attempt to insert an instruction in a network location where it will not fit, the system returns an **`Illegal operation`** message on the bottom line of the LCD.

It is also legal to overwrite one instruction with another one with an equal number of nodes. For example, if the network contains a three-node instruction such as an **ADD** block, it can be overwritten by any other valid three-node instruction (a **SUB**, a **BLKM**, an **EMTH** instruction, etc.). However, the three-node **ADD** instruction cannot be

overwritten by any two-node instruction (**UCTR**, **T1.0**, **JSR**, etc.) or on-node instructions (**SKP**, **RET**, etc.).

☞ **Note** The system does not allow you to overwrite a one-node instruction—e.g., **SKP**—with a one-node element such as a contact or coil.

Placing an Instruction in a Network

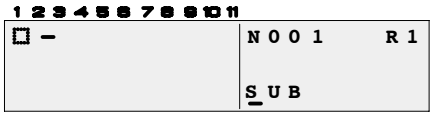
To enter a logic instruction, follow this nine-step procedure:

Step 1. Move the cursor to a legal top-node location in the logic display area of the LCD with fon ▲, foff ▼, disa ◄, and enab ► .

Step 2. Push the desired key sequence (see the table on the previous page) to reach the functional category for the desired instruction. For example, if you want to enter an **SUB** instruction, which is a math function, push ‡ math 6 .

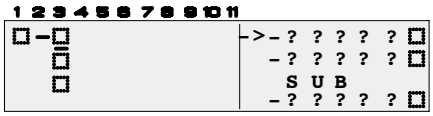
Result. As soon as you push ‡, a ‡ symbol appears in column 12 of the top line of the zoom screen, indicating that the next character entered from the keypad will be the superior (red) character. Once that character is entered, ‡ disappears.

The first in the list of available math instructions appears on the fourth line of the zoom display with the cursor under it:



Step 3. Use enab ► or disa ◄ to scroll to the desired instruction. In this example, scroll until **SUB** appears, then push command enter .

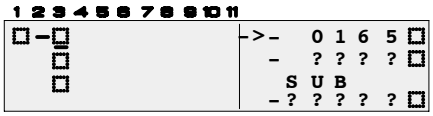
Result. The node symbols appear in the appropriate columns of the logic display area, with the cursor under the top node symbol. The reference data for the new instruction needs to be entered and appears in the zoom display as **?????**. For example:



Notice that the arrow in the zoom display on the right half of the screen points to the top node. This corresponds to the cursor location in the logic display on the left half of the screen.

☞ **Note** If you decide that you do not want to enter an instruction from this functional category in these node locations, scroll to **CANCEL** and push command enter .

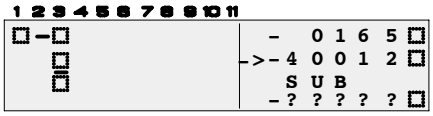
Step 4. Type the reference number you want in the top node, then push command enter . For example, if you enter the constant value **165** in the top node of the **ADD** block, the screen looks like this:



Step 5. Push command enter, then push foff ▼.

Result. The value in the top node is set, the cursor in the logic display moves to the lower node, and the arrow in the zoom display moves down to the line corresponding to the lower-node value.

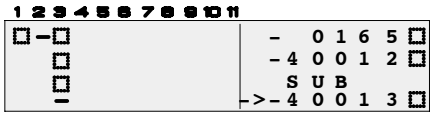
Step 6. Type the reference number you want to appear in the lower node of the instruction. For example, if you enter **40012** in the example above, the screen looks like this:



Step 7. Push command enter, then push foff ▼.

Result. The value in the lower node is set, the cursor in the logic display moves to the lowest node, and the arrow in the zoom display moves down to the line corresponding to the lowest-node value.

Step 8. Type the reference number you want to appear in the bottom node of the instruction. For example, if you enter **40013**, in the example above, the screen looks like this:



Step 9. Push command enter.

Result. The cursor effectivity returns to the logic display on the left side of the screen.

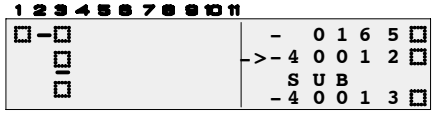
You have completed the setup of the new instruction. In this example, you have defined a three-high instruction that will subtract the contents of register 40012 from the constant 165 and will store the difference in register 40013.

Overwriting an Old Instruction with an New One

You can overwrite an existing instruction in a ladder logic network with a new instruction if the new instruction has the same number of nodes as the old one.

For example, if you want to insert an **ADD** instruction in the network locations where a **SUB** instruction currently resides, follow this procedure:

Step 1. Using fon , foff , disa , and enab , move the cursor under any one of the three **SUB** instruction nodes in the logic display. For example:

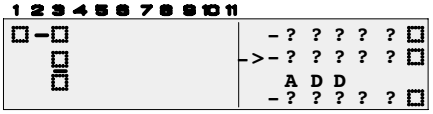


Step 2. Push ‡ math 6 to access the math instructions, scroll to **ADD** with enab or disa .

☞ **Note** If you decide that you want to overwrite the **SUB** instruction at this point in the procedure, scroll to **CANCEL** and push command enter .

Step 3. To overwrite the current **SUB** instruction, push command enter .

Result. The previous **SUB** reference data in the zoom display is replaced by **?????**'s:



Step 4. Use fon to move the cursor under the top node of the new **ADD** instruction, then follow steps 4 ... 9 of the previous procedure to enter the reference data into the nodes on the zoom screen.

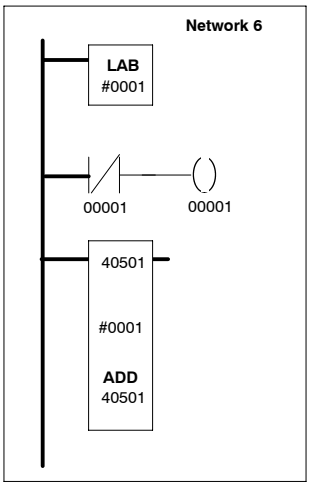
How the HHP Displays Ladder Logic Instructions

The symbols in the HHP's network display do not distinguish between nodes in a single-node element and nodes in a two- or three-high ladder logic instruction. To determine the context and content of a selected node, you need to view it in the zoom display.

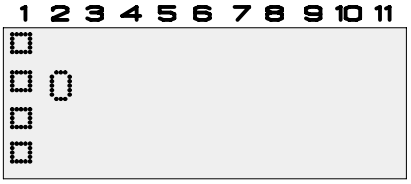
The zoom display handles ladder logic instructions differently than contact, coil, and short elements. The name of the instruction is usually shown in the third line of the zoom display, and the reference number or integer constant stored in that node is shown in the fourth line. If the instruction is two- or three-high, the references/values associated with the unselected node(s) are also shown in the zoom display.

A One-high Instruction

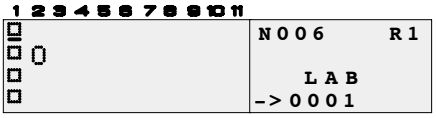
Suppose you are using an HHP to look at the following subroutine network:



The first four rows of this network (without power) would appear like this in the HHP's network display:



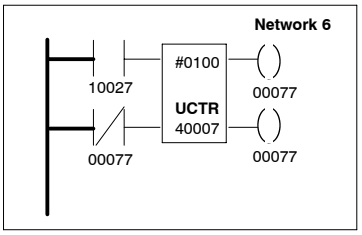
If you place the cursor in row 1, column 1, you select the one-high instruction **LAB**. The HHP screen display will look like this:



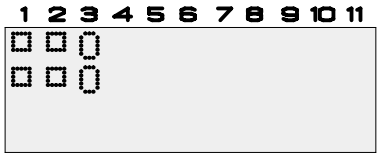
The top line of the zoom display tells you that the node is in network #6 (**N006**) and in row 1 (**R1**) of that network. The third and fourth lines give you the name and integer content of this one-high node. The arrow in the zoom display points to the value in the selected node.

A Two-high Instruction

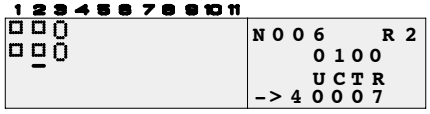
Suppose you are using an HHP to look at the following control network:



It would appear like this on the HHP's network display:



If you place the cursor in row 2, column 2, you select the bottom node of the two-high instruction **UCTR**. The HHP screen display will look like this:

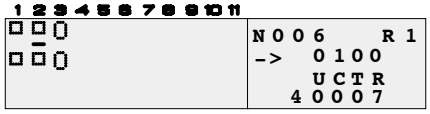


The top line of the zoom display tells you that the node is in network #6 (**N006**) and in row 2 (**R2**) of that network.

The second line in the zoom display shows the value in the top node of the **UCTR** instruction block. Even though this node is not the one selected in the network display, the zoom display shows its contents.

The third and fourth lines give you the name and content of the bottom node of the **UCTR** instruction. The arrow on the fourth line points to the value in the cursor-selected node.

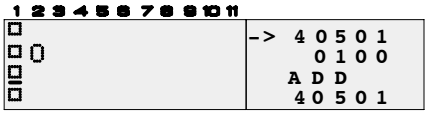
If you move the cursor up to row 1, column 2 of the network display—i.e., if you select the top node of the **UCTR** block—the following screen appears:



In the zoom display, the arrow has moved to the second line, indicating that a new node value has been selected, and the row number changes (to **R1**) to indicate the new cursor position.

A Three-high Instruction

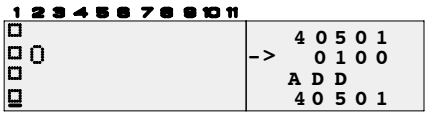
Suppose we look again at the subroutine network shown on the previous page, and this time we move the cursor down to row 3 of column 1. The HHP screen would then look like this:



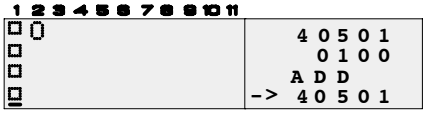
All four lines of the zoom display are used to show the name and node contents of a three-high instruction. The arrow points to the register value of the top node, which is the node selected in the network display.

Notice that all the instruction's node contents are shown in the zoom display, even though the bottom node is not shown in the network display. Notice also that because a three-high instruction requires all four lines in the zoom display, the network and row numbers are not given.

If you move the cursor down to row 4, column 1, the following screen appears with the cursor in the network display on the selected (middle) node of the **ADD** instruction and the arrow in the zoom display pointing to the value of that node:



If you move the cursor down one more row in the network display, the following screen appears:



The arrow in the zoom display shows that you have selected the bottom node of the **ADD** instruction.

Notice that the network display symbols on the left have changed dynamically—they have shifted up by one row. At this point, the network display is showing rows 2 ... 5 of the logic network instead of rows 1 ... 4 shown previously.

Errors of Range and Type

As you enter register references or constants into the zoom display in logic edit mode, the HHP will notify you if you enter an illegal reference number. A node reference can be illegal if it is of the wrong reference type or if its magnitude is out of range.

If one of these errors occurs in logic editing, a generic message appears on line 4 of the LCD.

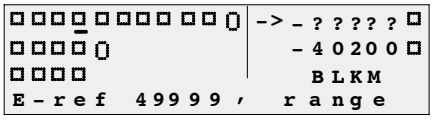
Range Errors

Both register numbers and constant values can be illegal because of range errors. A register is out of range when its magnitude—i.e., the value of its four trailing digits—exceeds the maximum value allowed for that reference type.

The default maximum limit is a function of reference type and PLC model:

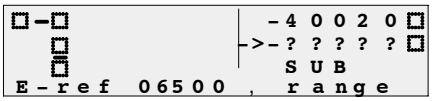
Reference Type	Default Range	
	311 / 411	512 / 612
0*x*	00001 ... 01024	00001 ... 01536
1*x*	10001 ... 10256	10001 ... 10512
3*x*	30001 ... 30032	30001 ... 30048
4*x*	40001 ... 40400	40001 ... 41872

In the sample screen below, the cursor is under the top node of a **BLKM** instruction. The 4*x* reference number we attempt to type into that top node is an out-of-range register reference of **49999**. The HHP has posted an **E-RANGE** error message on the bottom line to indicate this error:



Many instructions allow or require the use of a constant value in one or more of their nodes. The legal range for constants varies in different instructions. Refer to the instruction descriptions in the ***Modicon Micro Ladder Logic Manual*** (GM-MICR-LDR) for more details.

In the sample screen below, the cursor is under the middle node of a **SUB** instruction. If we attempt to enter a constant value of 6500 into the middle node, the following message appears on the bottom line of the screen:

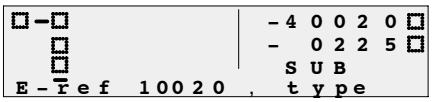


The maximum value you can assign to a constant in the middle node of a **SUB** instruction is 999.

Data Type Errors

The nodes in many instructions are limited to certain types of register references. Refer to the instruction descriptions in the ***Modicon Micro Ladder Logic Manual*** (GM-MICR-LDR) for more details.

For example, the bottom node of a **SUB** instruction must be a 4*x* holding register. If you attempt to enter a different reference type—a 0*x*, 1*x*, or 3*x* reference—a *type* error message appears on the bottom line of the screen:



Special Commands for Editing Logic

A set of seven special logic editing commands can be accessed by pushing ‡ command enter .

Command Descriptions

These commands do the following:

- **LMARK**—Ties the ON/OFF state of one discrete input or output point to the LED on the HHP
- **OPEN ROW**—Shifts all node data currently on and below the selected row in a ladder logic network down by one row
- **OPEN COL**—Shifts all node data on and to the right of a selected column in ladder logic to the right by one column
- **CLOSE ROW**—Forces all node data currently on and below an empty row in a ladder logic network up by one row
- **CLOSE COL**—Forces all node data currently on and to the right of an empty column in a ladder logic network left by one column
- **DEL NET**—Deletes the current network and all of its nodal contents from the ladder logic program
- **AVAIL?**—Tells you how many words are still available for use in user program memory and how much has already been used
- **CANCEL**—None of the above

How to *LMARK* a Discrete

You can select one discrete input (1*x* reference) or coil (0*x* reference) to have its state monitored by the HHP. Whenever the state of the selected discrete is detected as ON, the **status** LED on the HHP goes ON.

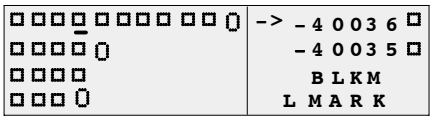
No more than one discrete reference in the logic program can be LMARKed at any given time.

☞ **Note** The HHP monitors the ladder logic program several times per second. However, if the LMARKed discrete is changing states more frequently than the HHP is monitoring it, then all of its state changes will not be detected.

To LMARK a Discrete Reference

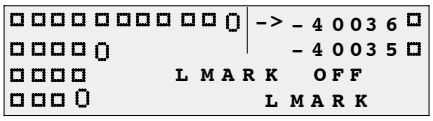
Step 1. Push ‡ command enter .

Result. The text string **LMARK** should appear on the bottom line of the zoom display on the HHP. If a different command name appears there, scroll through the list of commands with disa ◄ or enab ► until **LMARK** appears:



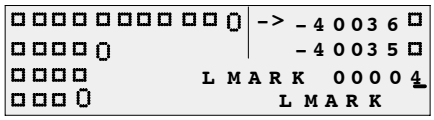
Step 2. Push command enter .

Result. If no discrete reference is currently LMARKED, the following message appears on the screen (with the cursor in the last position in the bottom line):



If a discrete currently is LMARKed, the message at the bottom of the screen shows its

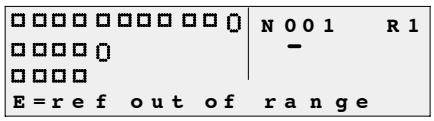
reference number. For example, if coil 00004 is LMARKed, the message reads:



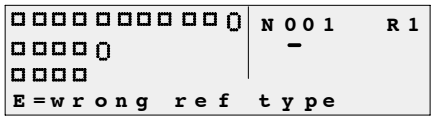
Step 3. Type the reference number of the discrete you want to monitor, then push command enter .

Result. The ON/OFF states of new reference will be monitored by the HHP. If another reference was previously LMARKed, it will no longer be monitored.

If you attempt to type in a 0*x* or 1*x* reference number whose magnitude is too big—i.e., 0*x* > 1024 in a 311/411, > 1536 in a 512/612; 1*x* > 256 in a 311/411, > 512 in a 512/612—the HHP will display a range error message at the bottom of the screen:



If you attempt to type in a non-discrete reference type—i.e., any leading digit other that 0 or 1—the HHP will display a reference-type error message at the bottom of the screen:



In order to return to an **LMARK** OFF condition, type **00000** as the reference number.

LMARK remains in effect when you switch back and forth between logic edit and logic data modes.

Opening and Closing Rows in a Network

The HHP allows you to shift rows of logic nodes up or down within a network in logic edit mode. The logic that is shifted in an **OPEN**/**CLOSE ROW** command moves as a block, maintaining consistent topological relationships.

The **OPEN ROW** command shifts all the nodal logic on the row where the cursor is placed and all the nodal logic below it in that network down by one row. This leaves the row where the cursor was placed open—i.e., no logic is programmed into any nodes on that row.

The **CLOSE ROW** command shifts all the nodal logic below the empty row where the cursor is place up by one row to fill the empty row.

Limits

Remember that no nodal information can be deleted or moved to new networks using the **OPEN**/**CLOSE ROW** commands. Therefore the HHP disallows the opening or closing of rows under certain circumstances:

- If row 6 contains a vertical short, **OPEN ROW** is not allowed
- If row 7 of a network contains any logic nodes, **OPEN ROW** is not allowed in that network
- If the row where the cursor is placed contains any node of a multi-node instruction other than the top node, **OPEN ROW** is not allowed
- If the row in which the cursor is placed contains any nodal logic (including shorts), **CLOSE ROW** is not allowed in that network

How to Open a Row

Step 1. Place the cursor in any column of the row you want to open.

Step 2. Push ‡ command enter .

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **OPEN ROW** appears, then push command enter .

Result. All the ladder logic node symbols on and below the selected row in the logic display shift down by one row. The cursor remains positioned in the newly opened row.

☞ **Note** The **OPEN ROW** command is executed as soon as you push command enter ; no confirmation message is issued. If you want to undo an **OPEN ROW** command, issue a **CLOSE ROW** command with the cursor positioned in the same location.

How to Close a Row

Step 1. Place the cursor in any column of an empty row that you want to close.

Step 2. Push ‡ command enter .

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **CLOS ROW** appears, then push command enter .

Result. All the ladder logic node symbols below the selected row in the logic display shift up by one row. The cursor remains positioned in the newly closed row.

☞ **Note** The **CLOSE ROW** command is executed as soon as you push command enter ; no confirmation message is issued. If you want to undo a **CLOSE ROW** command, issue an **OPEN ROW** command with the cursor positioned in the same location.

Opening and Closing Columns in a Network

The HHP allows you to shift columns of logic nodes left or right within a network in logic edit mode. The logic that is shifted in an **OPEN**/**CLOSE COL** command moves as a block, maintaining consistent topological relationships.

The **OPEN COL** command shifts all the nodal logic in the column where the cursor is placed and all the nodal logic to the right of that column further to the right by one column. This leaves the column where the cursor was placed open—i.e., no logic is programmed into any nodes in that column.

The **CLOSE COL** command shifts left all the nodal logic to the right of an empty column.

Limits

Remember that no nodal information can be deleted or moved to new networks using the **OPEN**/**CLOSE COL** commands. Therefore the HHP disallows the opening or closing of columns under certain circumstances:

- If column 11 of a network contains any logic coils, **OPEN COL** is not allowed in that network
- If column 10 contains any non-coil logic nodes (including shorts), **OPEN COL** is not allowed in that network
- If the column in which the cursor is placed contains any nodal logic, **CLOSE COL** is not allowed in that network

How to Open a Column

Step 1. Place the cursor in a node of the column you want to open.

Step 2. Push ‡ command enter .

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **OPEN COL** appears, then push command enter .

Result. All the ladder logic node symbols in and to the right of the selected column shift to the right by one column. The cursor remains positioned in the newly opened column.

☞ **Note** The **OPEN COL** command is executed as soon as you push command enter ; no confirmation message is issued. If you want to undo an **OPEN COL** command, issue a **CLOSE COL** command with the cursor positioned in the same location.

How to Close a Column

Step 1. Place the cursor in a node of an empty column that you want to close.

Step 2. Push ‡ command enter .

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **CLOS COL** appears, then push command enter .

Result. All the ladder logic node symbols to the right of the selected column shift left by one column. The cursor remains positioned in the newly closed column.

☞ **Note** The **CLOSE COL** command is executed as soon as you push command enter ; no confirmation message is issued. If you want to undo a **CLOSE COL** command, issue an **OPEN COL** command with the cursor positioned in the same location.

Deleting a Network from the Ladder Logic Program

The ‡ del srch key sequence allows you to delete individual logic instructions and logic elements from a network, but it does not allow you to delete the network itself. Even if you delete all logic from the network nodes with ‡ del srch , the empty network will remain in ladder logic. The only way to delete the entire network is with the **DEL NET** command.

With **DEL NET**, you can delete the network and all of its nodal content in a single command. To delete a network, follow this procedure:

Step 1. Bring up the network you want to delete on the LCD of the Hand-held Programmer. You can page to the desired network in logic edit mode with botm next or top prev, or you can go directly to the desired network via back get.

Step 2. Push ‡ command enter.

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **DEL NET** appears, then push command enter.

Step 4. A confirmation message—**Confirm del net *n***—appears on the third line of the screen, where ***n*** is the number of the network you want to delete. Push command enter again to execute the **del net** operation.

Result. The network and all its logic content are deleted from the ladder logic program. All networks that previously followed the deleted network in the logic flow are renumbered—e.g., if network 3 of a five-network program is deleted, networks 4 and 5 become networks 3 and 4, respectively.

Caution Once you have issued and confirmed a DEL NET command, it cannot be undone. The logic in the deleted network can not be recovered.

Checking for Available User Logic Memory

You can check the number of words of user logic memory while you are in logic edit mode with the **AVAIL?** command. To access this information, follow this procedure:

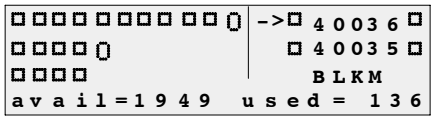
Step 1. From any network in logic edit mode, push ‡ command enter.

Result. A text string, probably **LMARK**, appear on the bottom line of the zoom display on the HHP.

Step 3. Scroll through the list of commands with disa ◄ or enab ► until **AVAIL?** appears, then push command enter.

Result. A message appears on the bottom line of the zoom display, indicating the number of words of user logic that have already been used and the number of words still available.

For example, the message:



indicates that of a total 2085 words of user program memory, 136 words have been used to date and 1949 words are still available for logic programming

(or for storing loadable ladder logic instructions).

Escaping from the Command List

If you get into the command list and want to get back out without executing any of the commands described here, use the **CANCEL** selection. Simply use dlsa ◄ or enab ► to scroll to **CANCEL**, then push command enter . The list of commands will disappear and the screen will return to the logic edit display.

Searching Nodes for Program Data

The [del srch] and [‡] [trac frmt] key sequences provide you with several node search capabilities in logic edit mode. You can search:

- Sequentially for all occurrences of a reference number (both explicit and implied) in ladder logic
- Specifically for a discrete (0*x* or 1*x*) reference number assigned to a logic element—e.g., normal coil 00001, N.O. contact 10004, negative transitional contact 10005
- Sequentially for all disabled coils in ladder logic
- Specifically for the individual coil tied to a selected contact in ladder logic
- Specifically for the subroutine **LAB** instruction associated with a selected **JSR** instruction in ladder logic

☞ **Note** You must search for an entity with a reference number assigned to it. You cannot search for a constant value or for an unspecified logic element type.

Starting a Search

All search operations must proceed in a forward direction from the starting point in the ladder logic program. You cannot search backward for previous occurrences of an element or reference, and you cannot wrap the search process from the end of logic back to the beginning.

To initiate a search starting at the current cursor position in logic, push [botm next] (in the top right corner of the HHP keypad). To initiate a search starting at the row 1 column 1 position in ladder logic network 1, push [back get] (on the top left side of the HHP keypad).

Once the search has been initiated, it will proceed forward through the logic program (in the direction of the logic scan) until it encounters either the element being searched for or the end of logic. To continue the search after an instance of the reference has been found, push [del srch], then [botm next].

Ending a Search

Each time a search is successfully completed, it is effectively ended. To return to the starting point, push [‡] [back get].

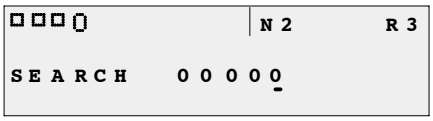
To cancel out of search mode, enter the value **00000** and push [clr exit]. The **SEARCH** message disappears, and the HHP returns you to the cursor position you were at before you initiated the search process.

Searching for a Reference Number

You can search for a specific node reference in logic edit mode with the following procedure:

Step 1. Push [del srch].

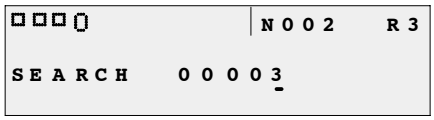
Result. A message appears on the third line of the LCD:



The cursor appears under the rightmost 0 in the reference number. Notice in this example screen that the current location in ladder logic is in row 3 of network 2.

Step 2. Type the reference number you want to search for.

For example, if you want to search for occurrences of coil 3 in ladder logic, type:



Step 3. If you want to initiate the search

at the current logic location—**N002**, **R3**—push botm next .

If you want to initiate the search

at the beginning of ladder logic, push back get .

Result. The HHP initiates the search from the indicated starting point.

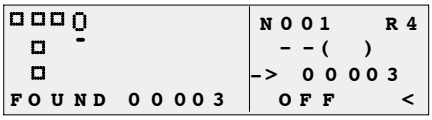
Step 4. When the HHP finds the next sequential occurrence of the reference it is searching for, it displays it on the LCD.

Result. The way in which the HHP displays the reference is determined by the kind of reference that has been found. The display will differ depending on whether the reference is explicit or implied, and whether it is a discrete or a register.

Finding Explicit References

An explicit reference is one that appears directly in the ladder logic display. It could be the discrete reference assigned to a contact or coil, or it could be a register that appears in a node in one of the instructions.

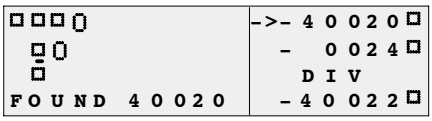
When a search finds an explicit discrete reference, a display similar to this appears:



where

- A message appears on the left side of the bottom line in the display stating that the specified reference has been found
- The cursor appears in the logic display under the found reference
- The arrow in the zoom display points to the reference number (on the third line)

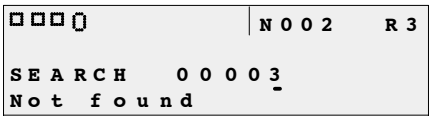
When a search finds an explicit register reference, a display similar to this appears:



where

- A message appears on the left side of the bottom line in the display stating that the specified reference has been found
- The cursor appears in the logic display under the found reference
- The arrow in the zoom display points to the reference number (in this case, on the first line to indicate that it is the top node of the **DIV** instruction)

If the HHP cannot find any occurrences of the specified reference, it returns a **Not found** message at the end of the search:

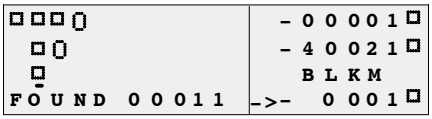


Finding Implied References

An implied reference is one that exists and can be used in the program but is not directly shown in the ladder logic display. Both register and discrete references can be implied in ladder logic.

For example, 16 discrete references can be stored in a 4*x* source register in a matrix instruction such as **BLKM**. The 4*x* reference may appear in the logic display, but the 16 discretes stored in the register are not shown—they are *implied*.

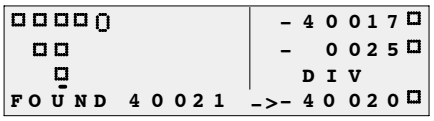
Say, for instance, you are searching for coil 00011 and the search reaches a **BLKM** where discrete references 00001 ... 00016 are being moved into holding register 40021. The result of the search would show a screen something like this:



The message on the bottom left of the screen tells you that an instance of discrete reference has been found.

A register reference can also be implied in ladder logic. In the **DIV** instruction, for example, the *explicit* 4*x* holding register that appears in the bottom node stores the quotient of the division operation while *implied* register 4*x* + 1 stores the remainder of the division.

Say, for instance, you are searching for register 40021 and the search reaches a **DIV** where register 40020 is displayed in the bottom node. The result of the search would show a screen something like this:



The message on the bottom left of the screen tells you that an instance of register reference 40021 has been found, and the arrow in the zoom display indicates that its position is implied behind register 40020.

☞ **Note** Whenever an implied (discrete or register) reference is found, the cursor in the logic display appears under the bottom node symbol of the instruction, and the arrow in the zoom screen points to the bottom node of the instruction. The implied reference itself is not necessarily contained in the bottom node of the instruction.

Special Search Capabilities for Discrete References

There are several ways in which you can limit a search to specific types of discrete references. You can limit the search to a specific type of coil or contact, or you can limit the search to disabled logic elements associated with discrete references in the logic program.

Finding Specific Discrete Logic Elements

When you type in the reference number of a discrete that you want to search for, you can further delimit the search by pushing one of the following keys:

Key Sequence	Discrete Element Type
‡ + A	N.O. contact
‡ + B	N.C. contact
‡ + 1 yes	Positive transitional contact
‡ + 2	Negative transitional contact
‡ + 3	Memory-retentive coil
‡ + C	Normal coil

For example, if you want to search for normally open (N.O.) contact 10008, you can invoke a search and type **10008** followed by ‡ A . The following appears on the screen:

```
□□□0                    N 2      R 3

SEARCH    1 0 0 0 8_    —| |—
```

When you push botm next or back get , the HHP searches for the next explicit occurrence of N.O. contact 10008. The search ignores any implied references to 10008 and any explicit occurrences of 10008 that are assigned to logic elements other than an N.O. contact.

If an instance of the specific reference you are searching for is found, a screen similar to this appears:

```
□□□0                    N 0 0 3    R 1
 -                        - -|   |
                        -> 1 0 0 0 8
FOUND 1 0 0 0 8          —| |—      <
```

The message on the bottom left of the screen tells you that the discrete you searched for has been found. The cursor in the logic display appears under the found contact, and the information in the zoom display tells you that it is an N.O. contact located in row 1 of network 3.

If the HHP does not find any instances of the kind of discrete you are searching for, a screen similar to this appears:

```
□□□0                    N 0 0 3    R 1

SEARCH    1 0 0 0 8_    —| |—
Not found
```

If the HHP finds an instance of the discrete you are searching for, you can continue to search for more instances of that discrete by pushing del srch , then botm next . You can continue to search for instances of the discrete using del srch , then botm next until you receive a **Not found** message. At this point, you know that there are no more instances of this particular discrete in the program after the currently found one.

☞ **Note** The ability to limit the search to a specific discrete reference type is available only for *explicit* discrete references in the logic program. It cannot be used in a register reference search or in a search for implied references.

Finding Disabled Discretes

The HHP can be used to search the logic program for disabled 0*x* or 1*x* references. You can search sequentially through the program for all instances of a disabled 0*x* or 1*x* references.

To initiate such a search:

Step 1. Push [del srch], then type the reference number you want to start with.

If you want to start the search at a specific disabled 0*x* or 1*x* reference, type it in; if you want to search sequentially for all disabled 0*x* (coil) references, type **00000**; if you want to search sequentially for all disabled 1*x* references, type **10000**.

Step 2. Push [‡] [disa ◄].

Result. The display of the discrete reference you are searching for contains the delimiter **DIS**, indicating that the search is restricted to instances of the disabled reference.

For example, if you want to search for all disabled coils in the logic program, the LCD would look like this:

```
□□□()                  N002    R3

SEARCH    00000_   DIS
```

If you want to start a search at a specific disabled discrete, say contact 10005, in the logic program, the LCD would look like this:

```
□□□()                  N002    R3

SEARCH    10005_   DIS
```

Step 3. Push [back get].

Result. The HHP starts searching the program at 10005 for the disabled discrete you have specified. If one is found, a screen similar to this appears:

```
□□□□()_                N001    R1
  □□                     --(   )
   □                  ->  00003
FOUND  00003  DIS                <
```

The message on the bottom left of the screen tells you that the disabled discrete you searched for has been found. The cursor in the logic display appears under the found coil (00003), and the information in the zoom display tells you that it is a normal

coil located in row 1 of network 1.

If the HHP does not find any disabled discretes of the kind you are searching for, a screen similar to this appears:

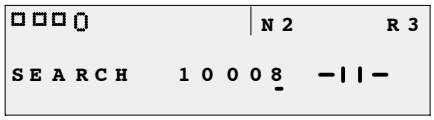
```
□□□()                  N002    R3

SEARCH    00001_   DIS
Not found
```

Canceling a Limited Search

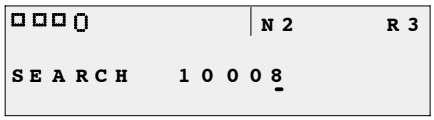
When you place a delimiter in the search command line, that delimiter will remain in place every time you push [del srch] until it is removed from the command line. To remove a delimiter from the search command line, push [‡] [del srch].

For example, suppose you want to search for register 40005, and when you push [del srch] a screen like this appears:



This screen indicates that the previous search issued on the HHP was limited to N.O. contact 10008.

You must remove the N.O. contact delimiter (**-| |-**) from the command line before you can search for a register reference. To do this, push ‡ del srch ; the delimiter is cleared from the command line:



Now you can key in the new reference number you want to search for, in this case **40005**, then push back get . The HHP now begins to search for the new register reference.

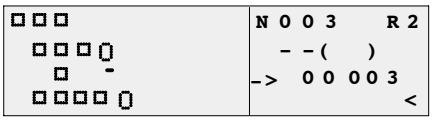
Tracing a Coil

The HHP also allows you to search for the specific coil that will fire a selected contact in ladder logic. This type of a search operation is known as *tracing*.

To trace a coil, place the cursor under a contact and push ‡ trac frmt . Since there can be no more than one coil traced in the logic, the botm next and back get keys are not used in a trace operation. A trace always begins in the first network of the logic program.

When the coil associated with the selected contact is found, the cursor appears under the traced coil in the logic screen and the arrow in the zoom screen points to the coil reference number on the third line.

For example, if the trace operation gives you this screen:



we see that the traced coil is coil 00003 and that it is located in row 2 of network 3.

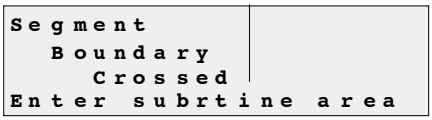
To return the HHP display to the original contact, push ‡ back get .

Finding a Subroutine

The ‡ subr data key sequence can be used in two ways:

- If you place the cursor on a **JSR** node in ladder logic and push ‡ subr data , the HHP will search for the **LAB** instruction associated with that **JSR** and position the cursor in that **LAB** node in the subroutine segment
- If you place the cursor on any node other than a **JSR** node in ladder logic and push ‡ subr data , the HHP will position you in the first network in the subroutine segment

If no networks exist in the subroutine segment, the HHP displays the following screen:



To return to the original network and cursor position, push ‡ back get .

Chapter 6
Monitoring and Editing Data Reference Values

- Logic Data Mode
- Reference Formats and Entry Fields
- Moving through the Logic Data Screen
- Editing Discrete Reference Data
- Editing Register Reference Data

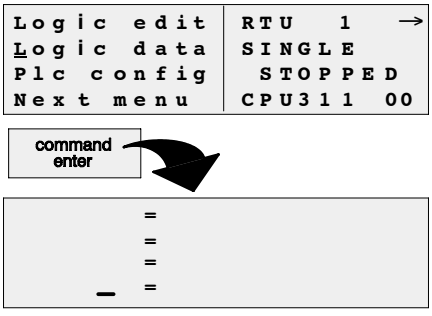
Logic Data Mode

In logic data mode, the HHP can monitor and edit the discrete and register reference data in the ladder logic program. To be able to get to logic data mode, the HHP must be connected to a PLC that is powered up and correctly configured.

Accessing Logic Data Mode

Logic data mode can be accessed from the top-level HHP screen or from a logic edit screen (see Chapter 5).

Use this screen/command sequence to access logic data mode from the main LCD screen:



To access the logic data mode from any screen in logic edit mode, push subr data .

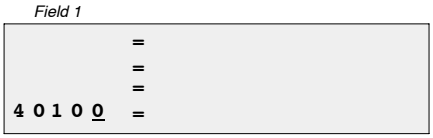
Entering Reference Data

When you enter logic data mode for the first time, the following screen appears:

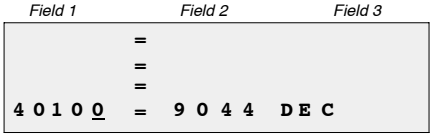
The cursor is positioned in column 5 on line 4. Column 5 is always the *field 1* cursor position for the data entries in the logic data screen.

To get the current *state* (in the case of a 0*x* or 1*x* discrete) or *value* (in the case of a 3*x* or 4*x* register) for any reference in PLC state RAM, enter the five-digit reference number with the cursor positioned in field 1.

For example, if you enter the holding register reference number 40100 into field 1:



then push command enter , the current value stored in that reference and the format in which it is displayed on the screen appear in fields 2 and 3:



In this example, we see that the current decimal value stored in holding register 40100 is 9044.

Similarly, if you enter a discrete reference number:

and then push command enter , the current state of that reference is displayed:

Here we see that coil 00001 is OFF and disabled. Whenever a coil is disabled, the condition is noted in field 3.

Note If field 3 of a specified discrete reference is empty, then that reference is enabled. In this screen:

```
          =
          =
          =
00001  =  OFF
```

coil 00001 is understood to be enabled.

Four reference data entries may be displayed per screen. To create an entry, push [fon ▲] or [foff ▼] to move from the current line to an empty line. The cursor will always move to column 5:

```
          =
          =
          =
40200  =  1255  DECIMAL
```

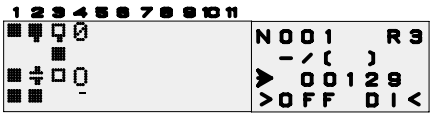
Then enter the desired reference number for that entry, then push [command enter]. Fields 2 and 3 will be filled in appropriately:

```
          =
          =
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

How to *Get* Reference Data from the Logic Edit Screen

If you are accessing the logic data screen from a logic edit screen and you are interested in seeing the data related to a specific logic node, you can do this with a combination of the [subr data] and [back get] keys.

For example, say your cursor in logic edit screen is under coil 00120:



Toggle to data reference mode by pushing [subr data], and the data screen appears as follows:

```
          =
          =
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

To display the coil 00120 reference data, simply move the cursor to line 2 with [fon ▲], then push [back get]. The following data entry appears:

```
          =
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

If all four lines are full in the logic data screen when you want to *get* reference data, you must delete one of the entries before executing [back get]. To delete an entry, place the cursor on the reference number (field 1), type **0**, then push [command enter].

All information in the entry will be cleared, and only the = sign will remain on the line. Then simply push [back get] to display the desired reference data item.

Reference Formats and Entry Fields

Each entry in a logic data screen is divided into three fields that describe the data in various user-selectable formats. As you can see by the examples presented so far, the data formats displayed in the logic data screen depend upon whether the specified reference is a discrete (0*x* or 1*x*) or a register (3*x* or 4*x*).

Discrete Formats

Discrete references can be displayed in *state* formats or, in some cases, in *matrix* formats.

State Formats

Any discrete reference can be displayed in a format that indicates its current state (ON/OFF). If the discrete is enabled, only two fields appear in that entry—the reference number and its ON/OFF state. If the discrete is disabled, a third data field appears to indicate that condition.

☞ **Note** If the third field in a discrete entry is empty, then the reference is enabled.

Two discrete references are shown in their state formats in the example below. The entry on line 3 is enabled—it uses two fields. The entry on line 4 is disabled—it uses three fields.

Field 1 *Field 2* *Field 3*

```
          =
          =
02047     =  ON
00001     =  OFF    DISABLE
```

Matrix Formats for Discrete Data

There are three additional formats in which a discrete may be displayed if and only if the discrete reference number falls on a 16-word boundary—e.g., 00001, 10017. These formats are called *matrix formats*.

By default, any discrete reference is displayed in its state format initially. If the discrete reference number falls on a word boundary, you can cycle through the following series of format displays using the trac frmt key:

- A *hex matrix* representing the states of the selected reference and the next 15 contiguous references
- A *high-byte binary matrix* representing the states (0 = OFF, 1 = ON) of the selected reference and the next seven contiguous references—e.g., if the selected discrete is 00017, then the high-byte matrix will show the states of references 00017 ... 00024
- A *low-byte binary matrix* representing the states (0 = OFF, 1 = ON) of the eight references that immediately follow the low-byte matrix—e.g., if the selected discrete is 00017, then the low-byte matrix will show the states of references 00025 ... 00032

When a matrix format is displayed for a discrete reference, the enabled/disabled condition of the reference states is not displayed. Suppose we take the logic data screen just shown and move the cursor under coil reference 00001.

Push trac frmt. The screen changes to something like:

```
          =
          =
02047     =  ON
00001     =  9B35  HEX
```

where the number **9B35** is the hexadecimal representation of the bit states of references 00001 ... 00016.

Push trac frmt again to produce:

```
          =
          =
02047     =  ON
00001     =  10011011  HIB
```

where the high-byte (**HIB**) binary value indicates the following states:

Reference	State
00001	ON
00002	OFF
00003	OFF
00004	ON
00005	ON
00006	OFF
00007	ON
00008	ON

Push trac frmt again to produce:

```
          =
          =
02047     =  ON
00001     =  00110101  LOB
```

where the low-byte (**LOB**) binary value indicates the following states:

Reference	State
00009	OFF
00010	OFF
00011	ON
00012	ON
00013	OFF
00014	ON
00015	OFF
00016	ON

Push trac frmt one more time to cycle the display back to state format.

If you are displaying a discrete reference on the logic data screen that does not fall on a word boundary, then pushing trac frmt has no effect.

Register Formats

A register reference entry on a logic data screen is always presented in three fields:

Field 1	*Field 2*	*Field 3*
Reference number	Reference value	Format type

Register values may be displayed in any one of four format types:

- Decimal
- Hexadecimal
- ASCII
- Binary (high byte, then low byte)

To sequence through these reference formats, place the cursor on the desired line in either field 1 or field 2 and push trac frmt. Each trac frmt sequences the format type displayed in field 3 and changes the way the reference value is displayed in field 2.

In the following example, the register reference on line 3 is in decimal format:

```
          =
          =
40100     =  9044  DECIMAL
40200     =  1255  DECIMAL
```

where the 16-bit value stored in register 40100 is displayed as **9044 DECIMAL**.

To change the display format of this value to ASCII, position the cursor as shown above and push trac frmt until the word **ASCII** appears in field 3. The screen entry will look like this:

```
Field 1          Field 2          Field 3
          =
          =
40100     =  #T      ASCII
40200     =  1255    DECIMAL
```

The trac frmt key can be used similarly to change the format to hexadecimal:

```
          =
          =
40100     =  2354    HEX
40200     =  1255    DECIMAL
```

where the value of the 16-bit value in register 40100 is displayed in hexadecimal format as **2354 HEX**.

The trac frmt key also changes the format to high-byte and low-byte displays of the binary representation of the register value. First, the high byte:

```
          =
          =
40100     =  00100011  HIB
40200     =  1255    DECIMAL
```

where the high byte of the 16-bit value in register 40100 is displayed in binary format as **00100011 HIB**.

Then the low byte:

```
          =
          =
40100     =  01010100  LOB
40200     =  1255    DECIMAL
```

where the low byte of the 16-bit value in register 40100 is displayed in binary format as **01010100 LOB**.

Moving through the Logic Data Screen

References and data in fields 1 and 2 of a line on the logic data screen can be edited by placing the cursor in the desired field, then entering values and/or issuing commands via the HHP keyboard. The ways in which the cursor moves within and between fields is determined by the format in which the reference data is displayed.

You cannot place the cursor in field 3. Changes to field 3 are implemented while the cursor is in field 1 or 2 of that entry, as described in the last section.

Moving between Line Entries

Use fon and foff to move the cursor up and down between lines in a logic data screen.

For example, if the cursor is on line 1 and you push fon, the cursor will wrap to line 4:

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

fon

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

If the cursor is on line 4 and you push foff, the cursor will wrap to line 1:

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

foff

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

or

```
       =
       =
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

foff

```
       =
       =
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

The < next > and < prev > Keys

Pushing botm next with the cursor placed in field 1 of an entry in the logic data screen scrolls the current entry up one line and replaces the vacated line with the next contiguous reference number, its value or state, and its format.

For example, if the cursor is on line 3:

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

the display changes as follows when botm next is pushed:

```
00120  =  OFF   DISABLE
40100  =  2354  HEX
40101  =  2355  HEX
40200  =  1255  DECIMAL
```

The entries previously on lines 2 and 3 are now moved up to lines 1 and 2, respectively, and the next contiguous reference—register 40101—now appears on line 3. The reference that previously appeared on line 1 has scrolled off the top of the display and now cannot be recovered.

Pushing top prev with the cursor placed in field 1 or field 2 of an entry scrolls the current entry down one line and replaces the vacated line with the previous contiguous reference number, its value or state, and its format.

For example, if the cursor is on line 1:

```
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
40200  =  1255  DECIMAL
```

the display changes as follows when top prev is pushed:

```
02046  =  OFF
02047  =  ON
00120  =  OFF   DISABLE
40100  =  2354  HEX
```

The entries previously on lines 1, 2, and 3 are now moved down to lines 2, 3, and 4, respectively, and the previous contiguous reference—coil 02046—now appears on line 1. The reference that appeared on line 4 of the previous screen has scrolled off the bottom of the display and now can not be recovered.

Editing Discrete Reference Data

The state of a discrete reference can be *forced* ON or OFF only when the discrete is disabled. If you try to change the state of an enabled discrete reference, the HHP will ignore the attempt.

Disabling an Enabled Discrete

Step 1. Display the desired reference in the logic data screen

Step 2. Place the cursor in either field 1 or 2 of the entry and push ‡ dlsa ◀ .

Result The text string **DISABLE** appears in field 3 of the entry, indicating that the discrete has been disabled

Changing the State of a Disabled Discrete

Step 1. Display the desired reference in the logic data screen

Step 2. Place the cursor in either field 1 or 2 of the entry

Step 3a. To force a currently ON reference to OFF, push ‡ foff ▼

Step 3b. To force a currently OFF reference to ON, push ‡ fon ▲ .

Result The previous ON/OFF status of the selected reference, displayed in field 2, is changed; the text string **DISABLE** remains in field 3 of the entry, indicating that the discrete is still disabled.

Enabling a Disabled Discrete

Step 1. Display the desired reference in the logic data screen

Step 2. Place the cursor in either field 1 or 2 of the entry and push ‡ enab ▶ .

Result Field 3 of the entry is empty, indicating that the discrete has been enabled.

Editing Register Reference Data

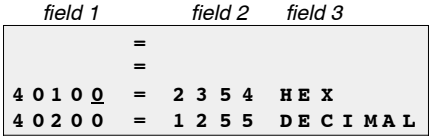
The way in which reference data in a 3*x* or 4*x* register can be edited depends upon the format in which it is displayed. The content of a 3*x* or 4*x* register can be displayed in any one of four formats—decimal, hexadecimal, ASCII, or binary.

Selecting a Format

As we have already seen, you can change the display format of register data with the trac frmt key. When the format changes, the data representation in field 2 and the format label in field 3 change to suit the selected format.

Editing Decimal and Hexadecimal Data

When register data is displayed in decimal or hexadecimal format, it appears on the screen like this:



where the register number is shown in field 1, the data in field 2, and the format labels in field 3.

To edit the data in field 2 of the desired entry, push enab to move the cursor into field 2. The cursor will appear under the leftmost character in field 2. If necessary, use fon or foff to move the cursor to the desired line in field 2.

As soon as you begin to enter new data in field 2, three things happen:

- The cursor moves to the rightmost position in field 2 with the newly entered character above it
- All other values on this line in field 2 are cleared to 0 (in hex format) or blank (in decimal format)
- All data throughout the screen is frozen

Decimal format allows an entry of five digits in the range 0 ... 65,535. Numbers greater than 65,535 are converted to (n – 65,535) with no leading zeros.

The new data is entered into the field from right to left. For example, if you want to replace an old hexadecimal value of 2354 with a new hex value of 4752, the line entry would look like this as you type in the new values:

Keystroke	LCD Display
enab ▶	40100 = 2354 HEX
cntr 4	40100 = 0004 HEX
move 7	40100 = 0047 HEX
timr 5	40100 = 0475 HEX
INI 2	40100 = 4752 HEX

To confirm the new value and unfreeze the data screen, push command enter. The PLC writes the changed data content to register 40100 and immediately begins to monitor and update all screen references.

Editing ASCII Data

The HHP keypad does not support the direct entry of new ASCII characters into the display because of the character limits on the keypad. You can, however, indirectly edit ASCII characters by using another format—e.g., hexadecimal—to represent the desired characters, then toggle the format to **ASCII**.

For example, if register 40100 contains the ASCII characters **#T**, and you want to change those characters to **GR**, follow these three steps:

Step 1. Place the cursor on the **40100** in field 1 and push trac frmt until the format changes to **HEX**.

Result. The data in field 2 changes from **#T** to **2354** (the hex equivalent of the ASCII characters).

Step 2. Edit the hex value, changing it from **2354** to **4752** (the hex equivalent of the ASCII characters **GR**).

Step 3. Using the trac frmt key, change the reference display format back to **ASCII**.

Result. The data content in field 2 changes from **4752** to **GR**.

Editing Binary Data

The binary representation of data in a 3*x* or 4*x* register can be shown in two format displays—one for the high byte (**HIB**) and one for the low byte (**LOB**). The hexadecimal value **2354**, for example, can be shown as:

Hexadecimal	Binary
23	00010111 HIB
54	00110110 LOB

Here is a procedural example that shows how you can edit the values in the **HIB** and **LOB** of register 40100, changing the the register value from 00010111, 00110110 (2354 hex) to 00101111, 00110100 (4752 hex):

Step 1. Set up your LCD screen so that the **HIB** and **LOB** displays for register 40100 are positioned one above the other:

```
          =
          =
40100  =  0 0 0 1 0 1 1 1  HI B
40100  =  0 0 1 1 0 1 1 0  LO B
```

Step 2. Push enab ▶ to move the cursor from field 1 to field 2 of the **HIB** display:

```
          =
          =
40100  =  0 0 0 1 0 1 1 1  HI B
40100  =  0 0 1 1 0 1 1 0  LO B
```

Result. The cursor moves under the leftmost **HIB** character in field 2.

Step 3. You must overstrike the leftmost character, even if you do not want to change it. In this case, simply type a **0**.

Result. The leftmost value remains **0** and the cursor moves right one position.

```
          =
          =
40100  =  0 0 0 1 0 1 1 1  HI B
40100  =  0 0 1 1 0 1 1 0  LO B
```

Step 4. Now use the enab ▶ key to move the cursor under the first character you want to change, in this case the third **0** from the left, and type in the desired value.

Result. The old value in the selected position will be overwritten by the new one.

Step 5. Repeat **step 4** for each bit value that you want to change in the **HIB** display. When all the changes have been made on this line, push command enter .

Result. The PLC writes the new values to the high byte of register 40100.

```
          =
          =
40100  =  0 0 1 0 1 1 1 1  HI B
40100  =  0 0 1 1 0 1 1 0  LO B
```

Step 6. Push foff ▼ to move the cursor down to the **LOB** display and repeat **steps 3 ... 5.**

Result. The PLC writes the new values to the low byte of register 40100.

```
          =
          =
40100  =  0 0 1 0 1 1 1 1  HI B
40100  =  0 0 1 1 0 1 0 0  LO B
```

Chapter 7
Using the HHP as a Data Transfer Device

- Modes of Data Transfer
- Some Important Cross-transfer Considerations
- A PLC-to-HHP Data Transfer
- An HHP-to-PLC Data Transfer
- Comparing HHP and PLC Data
- Saving a Logic Program to PLC Flash
- Hardware Requirements for Slave Mode Operations
- Uploading and Downloading Logic with an HHP in Slave Mode
- Changing HHP Executive Firmware

Modes Of Data Transfer

There are several situations in which a Hand-held programmer can be used as a data transfer device. These situations fall into two broad categories:

- The HHP and the Micro PLC to which it is connected transfer and/or compare data
- The HHP is used as a slave device to a non-PLC master device—e.g., to a personal computer running Modsoft Lite

The operations available under the first of these two categories are known as *transfer/save mode* operations. The operations available under the second category are called *slave HHP* mode operations. The operations available under these two modes will be discussed in detail in this and the next chapter.

Transfer/Save Mode Options

Four types of HHP-PLC data exchanges can be initiated by selecting menu options available in transfer/save mode. These options are:

- Transferring a controller environment —i.e., ladder logic program, PLC configuration, I/O map, etc.—from the PLC to the HHP
- Transferring a controller environment from the HHP to a PLC
- Verifying an actual PLC environment with one stored in the HHP and displaying the differences
- Initiating a save-to-PLC-flash operation in the PLC

These four transfer options are described in this chapter.

Slave HHP Mode Options

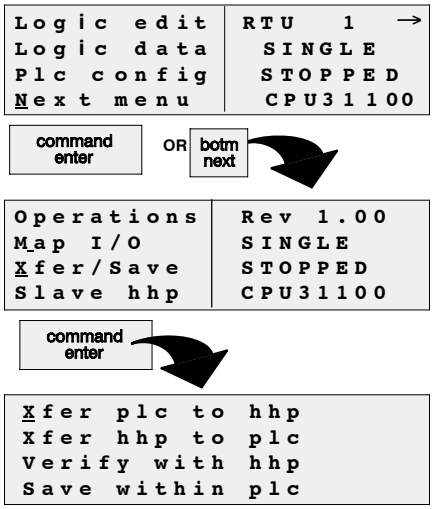
When the HP is a slave to a computer master device, the computer can use the HHP to:

- Download a ladder logic program from the computer to the HHP's flash
- Upload a ladder logic program from the HHP's flash to the computer
- Download a new HHP executive from the computer into the HHP's flash

These three HHP slave operations are described in this chapter. A fourth slave operation—the use of the HHP as a message entry/display panel—is described in Chapter 8.

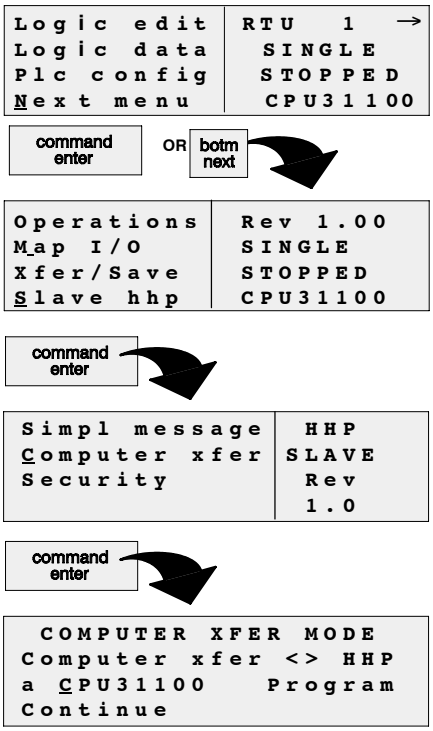
How to Access Transfer/ Save Mode

The transfer/save mode can be accessed in an HHP via the following menu path:



How to Access Slave HHP Mode

The slave HHP mode can be accessed in an HHP via the following menu path:



Some Important Cross-transfer Considerations

When you are using an HHP to transfer ladder logic programs from one PLC to another, or from/to another panel software such as Modsoft Lite, several compatibility issues need to be kept in mind.

Compatibility between Micro PLC Models

Because of the way the HHP reads and writes the PLC memory structure, it can support data transfers (logic program, I/O map table, and config table) between all CPU512 and CPU 612 models and between all CPU311 and CPU411 models.

However, it *does not* support data transfers between CPU311 and CPU512 models, CPU311 and CPU612 models, CPU411 and CPU512 models, or CPU411 and CPU612 models.

Editing the Target I/O Map after a Data Transfer

When you complete a legal data transfer from a source Micro PLC of one model type to a target Micro PLC of another model type, you may need to edit the I/O map in the target PLC.

For example, if you download a program to a 110CPU31100 (which has 24 VDC inputs and 12 relay outputs) to a 110CPU31101 (which has 115 VAC inputs, eight triac outputs, and four relay outputs), you will need to edit the **MIC** number that appears in LOC01 of the target PLC's I/O map.

Similarly, if you download a program created for a CPU612 PLC (which supports fixed analog I/O) to a CPU512, you may need to clear the **MIC** number that appears in LOC04 of the target PLC's I/O map.

Transferring Data Created on another Panel to the HHP

If you create a PLC environment with another panel software such as Modsoft or Modsoft Lite, you can do a couple of things that cannot be managed if that data is transferred to an HHP.

Limiting the Number Modules in a PLC Configuration

Software packages such as Modsoft and Modsoft Lite allow you to customize the number of locations in the PLC's I/O map—the default in Modsoft and Modsoft Lite is one LOC.

The HHP, on the other hand, cannot adjust the size of the PLC's I/O map table. If you are using an HHP, the only way you can set the I/O map size is by the *autoconfig* process, in which default values are used (see pages 30 ... 34). The I/O map size cannot be changed otherwise via the HHP.

If a PLC configuration with a restricted number of I/O LOCs (less than five in a CPU311 or CPU411, less than 20 in a CPU512 or CPU612) is transferred to an HHP, the HHP does not have the ability to increase the number of LOCs.

When you are creating a program with another software package, you should always configure the maximum number of I/O LOCs before you download to the HHP.

Creating an Optimized Program

Software packages such as Modsoft and Modsoft Lite also permit you to *optimize* your ladder logic execution. When the program is running under Modsoft or Modsoft Lite in optimize mode, the logic is solved as efficiently as possible, but certain operations—e.g., editing logic are not permitted while the PLC is running.

The HHP does not support optimize mode functionality. If a program in optimize mode is transferred to the HHP, the HHP is unable to take the program out of optimize mode. Thus, you are unable to edit your logic while the PLC is running.

A PLC-to-HHP Data Transfer

The HHP is a handy tool for transporting a ladder logic program from one Micro PLC to another. You can easily transfer a program running in a PLC to the HHP's flash, then use the HHP to load the program to other PLCs. The PLC-to-HHP transfer can take place whether the the PLC is running or stopped.

Accessing the PLC-to-HHP Transfer Screens

To transfer a ladder logic program to the HHP from the PLC to which it is connected, go into the the transfer/save mode from the top level HHP menu, then place the cursor under the first of the four menu selections on that screen—**Xfer plc to hhp**. Then push command enter .

Here is that screen/command sequence:

```
Operations
Map I/O          SINGLE
Xfer/save        STOPPED
Slave hhp        CPU31100
```

command enter

```
Xfer plc to hhp
Xfer hhp to plc
Verify with hhp
Save within plc
```

command enter

```
XFER TO HHP THE
CPU51201 PROG
Continue
```

The model number of the PLC to which you are connected should always appear on the second line of the screen. For example, the screen directly above indicates that the HHP is connected to a 110CPU51201 PLC.

Initiating the Data Transfer

If you do not want to continue the PLC-to-HHP data transfer at this point, you can push clr exit . The HHP returns you to the previous transfer/save mode menu screen.

To proceed with the data transfer, push command enter . The following confirmation screen appears:

```
TRANSFER PLC TO HHP
     ADDRESS 00000
        DATA 00000
Start
```

The address and data fields on the second and third lines of the screen are always set to **00000** prior to initiating the transfer.

You may still cancel the transfer at this point by pushing clr exit . To start the transfer push command enter the data transfer proceeds from this point until all the current PLC data is transferred to the HHP.

☞ **Note** There is no cancel capability for the transfer process once it has started.

As the data transfer is occurring, address locations in PLC memory that are being transferred to the HHP appear in the address field on the second line of

the screen. The data contained in the displayed address locations appears in the data field on the third line of the screen.

When the transfer is completed, a screen similar to this appears with a completion message on the bottom line:

```
TRANSFER PLC TO HHP
     ADDRESS F1380
        DATA 00499
Xfer to hhp complete
```

☞ **Note** If you start the data transfer, you should push command enter only once. Pushing the key more than once will exit you from the screen as soon as the transfer process finished—whether the transfer is successful or not. When you push the key only once, a message will appear at the bottom of the screen at the end of the transfer telling you whether or not it has completed successfully.

After the transfer is complete, push clr exit to return to the previous transfer/save mode menu screen.

An HHP-to-PLC Data Transfer

A logic program that is stored in the HHP's flash can be transferred to a Micro PLC. To properly transfer data from the HHP to a PLC, these conditions must be true:

- The PLC must be stopped
- If the program in the HHP was written for a 311/411 model PLC, then the target PLC must be any type of 311/411 PLC
- If the program in the HHP was written for a 512/612 model PLC, then the target PLC must be a 512/612 PLC

☞ **Note** If the HHP contains a 612 program with fixed analog I/O programmed, the program can be transferred to a 512 PLC but it cannot be run on that PLC unless you go into the I/O map editor of the target PLC and change LOC04 (see Chapter 4).

Accessing the PLC-to-HHP Transfer Screens

To transfer a ladder logic program from the HHP to a PLC, go into the the transfer/save mode from the top level HHP menu, then place the cursor under the second of the four menu selections on that screen—**Xfer hhp to plc**.

Then push command enter .

Here is that screen/command sequence:

```
Operations          |
Map I/O             |  SINGLE
Xfer/save           |  STOPPED
Slave hhp           | CPU31100
```

command enter

```
Xfer plc to hhp
Xfer hhp to plc
Verify with hhp
Save within plc
```

command enter

```
XFER TO CPU51200 THE
CPU51201 PROG IN HHP
Continue
```

The model number of the target PLC appears on the first line of the screen, and the type of PLC on which the program in the HHP was created appears on the second line. For example, the screen directly above indicates that the HHP contains a program written created with a 110CPU51201 PLC and that this program will be transferred to a 110CPU51200 PLC.

Initiating the Data Transfer

If you do not want to continue the HHP-to-PLC data transfer at this point, you can push clr exit . The HHP returns you to the previous transfer/save mode menu screen.

To proceed with the data transfer, push command enter . The following confirmation screen appears:

```
TRANSFER HHP TO PLC
     ADDRESS 00000
        DATA 00000
Start
```

The address and data fields on the second and third lines of the screen are always set to **00000** prior to initiating the transfer.

You may still cancel the transfer at this point by pushing clr exit. To start the transfer push command enter the data transfer proceeds from this point until all the current data in the HHP's flash is transferred to the target PLC.

> **Note** There is no cancel capability for the transfer process once it has started.

As the data transfer is occurring, address locations in HHP flash that are being transferred to the PLC appear in the address field on the second line of the screen. The data contained in the displayed address locations appears in the data field on the third line of the screen.

When the transfer is completed, a screen similar to this appears with a completion message on the bottom line:

```
TRANSFER PLC TO HHP
     ADDRESS F1380
        DATA 00499
Xfer to plc complete
```

After the transfer is complete, push clr exit to return to the previous transfer/save mode menu screen.

Comparing HHP and PLC Data

After you have transferred a logic program from a PLC to the HHP's flash or from the HHP to a PLC, you can compare—or *verify*—the two programs and see whether there are any differences between them.

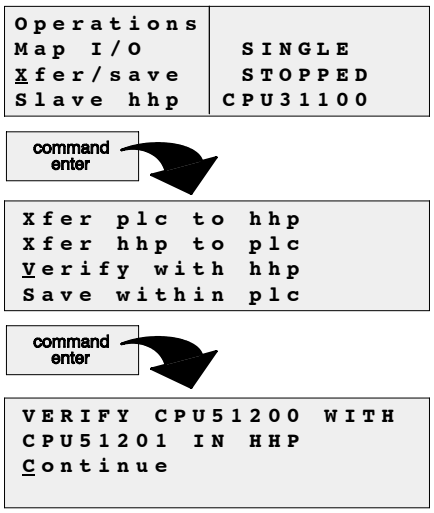
If the PLC involved in the transfer is not running at the time of the transfer and is not run before the comparison (verification) is made, then the two programs should match completely. If you have loaded a program to an HHP from a running PLC or if you put the PLC in run mode at any time after transfer and before the comparison is made, then miscompares will occur normally in the comparison process.

Accessing the HHP-PLC Verify Screens

To compare the ladder logic program in an HHP to the one in a PLC, go into the the transfer/save mode from the top level HHP menu, then place the cursor under the third of the four menu selections on that screen—**Verify with hhp**.

Then push command enter .

Here is that screen/command sequence:



The model number of the PLC appears on the first line of the screen, and the type of PLC for which the program in the HHP was created appears on the second line. For example, the screen directly above indicates that the program in the HHP was created for a 110CPU51201 PLC and the PLC in which the program resides is a model 110CPU51200.

Initiating the Program Verification

If you do not want to initiate the program verification at this point, you can push clr exit . The HHP returns you to the previous transfer/save mode menu screen.

To proceed with the program verification, push command enter . The following confirmation screen appears:

```
VERIFY ADDRESS 00000
       PLC DATA 00000
       HHP DATA 00000
Start/cont
```

The memory address on the top line and PLC and HHP data fields on the second and third lines of the screen are set to **00000** prior to initiating the verification operation.

You may still cancel the verification at this point by pushing clr exit. To initiate the verification, push command enter. The data verification begins. The data content of each address location in the PLC is compared to the same address location in the HHP program. The process proceeds uninterrupted unless or until a miscompare occurs.

The verification process stops if it finds a miscompare in any address location. A screen similar to this appears:

```
VERIFY ADDRESS 00067
       PLC DATA 08000
       HHP DATA 00000
Start/cont   **STOP**
```

In the sample screen above, we see that address 67 in the PLC contains a value of **08000** while address 67 in the HHP contains **00000**. If you want to continue with the verification process, push command enter; if you want to stop the process and return to the first transfer/save screen, push clr exit.

If you continue the verification process, or if the process completes itself without detecting any miscompares, the following screen appears:

```
VERIFY ADDRESS 00067
       PLC DATA 08000
       HHP DATA 00000
Start/cont   FINISHED
```

At this point, push clr exit to leave the verification screen.

What to Do if You Find an Unexpected Miscompare

Miscompares can occur normally if the PLC has been running at any point during or after the data transfer prior to initiating a verification process—dynamic events in logic program will change the content of some memory addresses as the program is scanned in run mode

If the PLC has been stopped constantly between the time the program was transferred and the verification process initiated, the data in the addresses of both the PLC and the HHP should be identical. If a miscompare is detected under this circumstance, you should proceed as follows:

Step 1. Exit from the current verification process and re-initiate a new verification process.

Step 2a If another miscompare occurs at the same location, exit from the verification screens and redo the data transfer between the PLC and HHP.

Step 2b If a miscompare occurs at a different location, check the physical connection between the HHP and PLC—the comm ports and the cable—for a condition that could cause a comm error.

If the source of the miscompares still cannot be identified, there may be a hardware problem in the PLC or in the HHP.

Saving a Logic Program to PLC Flash

The HHP can be used to issue a *save to flash* command in the Micro PLC to which it is connected. This command saves the program currently in the PLC to the PLC's flash memory. Once the program is stored in the PLC's flash, it cannot be lost during a power failure, even if the PLC does not use the battery or battery capacitor options.

Accessing the Save-to-Flash Screens

To initiate a save-to-flash command in the PLC from an HHP, go into the the transfer/save mode from the top level HHP menu, then place the cursor under the third of the four menu selections on that screen—**Save within plc**. Then push command enter .

Here is that screen/command sequence:

```
Operations
Map I/O          SINGLE
Xfer/save        STOPPED
Slave hhp        CPU31100
```

command enter

```
Xfer plc to hhp
Xfer hhp to plc
Verify with hhp
Save within plc
```

command enter

```
SAVE WITHIN PLC
POWER-UP RUN STATE
Will be Stopped
Start   saving
```

☞

To initiate the save, the PLC you are communicating with must be stopped. If the PLC is running when you attempt the save, a message appears on the bottom of the screen:

```
SAVE WITHIN PLC
POWER-UP RUN STATE
Will be Stopped
Controller  running
```

To stop the PLC, exit out of the transfer/save screens, select **Operations** on the top-level menu, and stop the PLC. You can then return to the transfer/save screens to initiate the save-to-flash operation.

Selecting a Start-up State for the PLC

The first item you are asked to specify on the **SAVE WITHIN PLC** screen is the power-up state of the program you are saving to flash. This condition is defined on the third line of the screen.

The default start-up state is **Stopped**. If the PLC adopts this ladder logic program in its power-up sequence, the PLC will come up in a stopped state. You will then need to go to the operations menu and select **Start** to put the PLC in run mode.

You may toggle the start-up state of the PLC to **Running** by pushing the disa ◄ or enab ► key. In this case, the PLC will come up running when it adopts the program in flash during its power-up sequence.

> **Note** The **Stopped** condition is generally considered the safer start-up condition, but you may consider powering up the PLC in a

Running condition in applications where the PLC is in a remote or inaccessible location.

You can still cancel the save-to-flash operation at this time by pushing clr exit. The HHP returns you to the first transfer/save screen.

Initiating the Save

When you have specified the desired start-up state for the program to be stored in the PLC's flash, push foff ▼ to move the cursor under the **Start saving** selection on the bottom line of the screen. To save the logic program to flash, push command enter.

If the PLC is stopped when you initiate the save, the message **Please wait** flashes on the bottom line of the screen until the operations completes. When the operation is completed, the following message appears on the bottom line of the screen:

```
SAVE WITHIN PLC
POWER-UP RUN STATE
Will be Stopped
SAVING SUCCESSFUL
```

Push clr exit to return to the first transfer/save screen.

Hardware Requirements for Slave Mode Operations

The HHP needs 150 mA from an external power source in order to operate. In slave mode operations where it is not connected to a Micro PLC, the HHP must draw its power from an independent external power source. All computer transfer operations require additional hardware to support the HHP in slave mode.

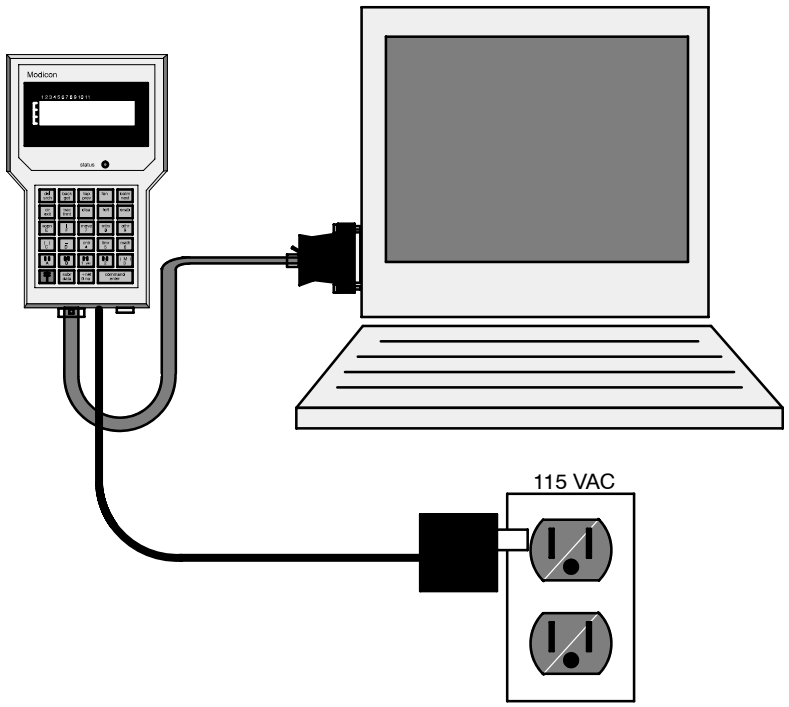
Modicon offers an HHP Computer Transfer Mode Kit (110VIA19200) to satisfy these special hardware requirements.

This kit includes three items:

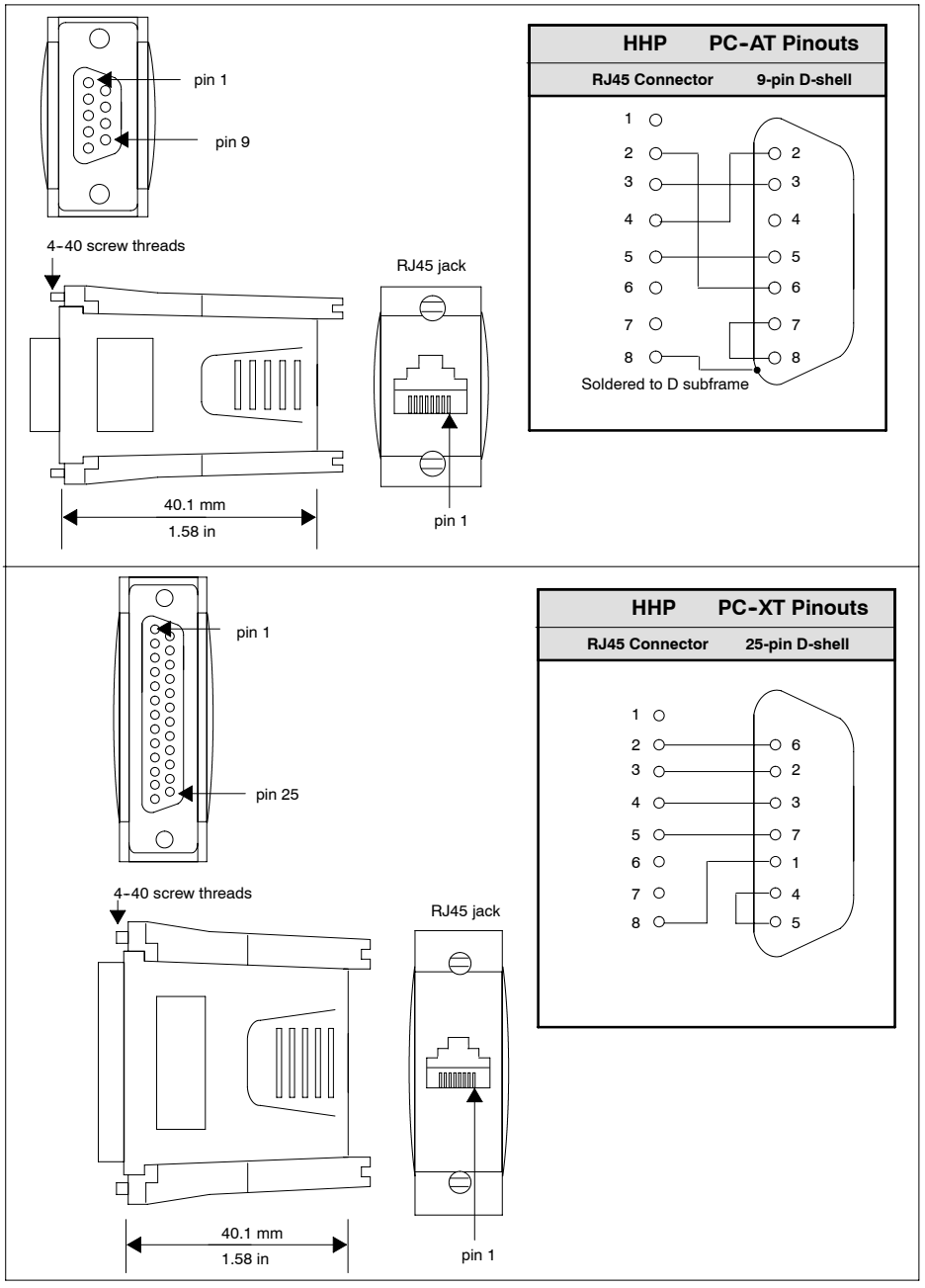
- A 115 VAC-to-DC power converter
- A 9-pin D-shell connector for using the HHP with an AT-type computer
- A 25-pin connector for using the HHP with an XT-type computer

The illustration below shows an HHP-computer connection. Plug one of the RJ45 connectors on the RS-232 cable into the HHP's communications port and the other one into the D-shell connector appropriate appropriate for your computer. The 2 m (78 in) power cord running from the AC-to-DC converter plugs into the power adapter connector on the bottom of the HHP.

Connecting an HHP to an Independent DC Power Source



The D-shell connectors provided in the 110VIA19200 kit have an RJ45 jack that allows them to clip directly onto the cable assembly. The illustrations below show the detail of the two D-shell connectors provided in the kit.



Uploading and Downloading Logic with an HHP in Slave Mode

With the HHP powered up in slave mode and connected to a computer running a panel software such as Modsoft Lite, you can load and copy user logic programs into and out of the HHP's flash.

In these operations, the computer running the panel software—i.e., the master device—issues the upload or download command. The HHP is semi-passive throughout the computer transfer—it simply accepts or transfers the data defined by the master.

The master treats the HHP as a Micro PLC in a computer transfer operation.

Powering Up the HHP

When the HHP is powered up from an independent power supply (as it must be when it operates in computer transfer mode), it will fail its start-up attempt to establish communications with a PLC. This situation is discussed in more detail in Chapter 2. The following top-level HHP screen appears when the communication attempt fails:

Hhp port	NO COMMS
Slave hhp	NO CONFIG
	UNKN PLC

Accessing Computer Transfer Mode

To access computer transfer mode, make the HHP a Modbus slave by moving the cursor under **Slave hhp** and push command enter .

Here is that screen/command sequence:

Hhp port	NO COMMS
Slave hhp	NO CONFIG
	UNKN PLC

command enter

Simpl message	HHP
Computer xfer	SLAVE
Security	Rev
	1.0

command enter

COMPUTER XFER MODE
Computer xfer <> HHP
a CPU31100 Program
Continue

The PLC model number that appears on the third line of the computer transfer screen tells the master device that the HHP holds a logic program obtained from a specific type of PLC. If the HHP does not currently have a logic program stored in its flash, the default PLC model number **CPU31100** appears, as shown in the sample screen directly above. If the HHP's flash does currently contain a logic program, then the PLC model number for which that program was written appears by default on the third line with an asterisk (*) next to it. The asterisk indicates that the HHP does contain a valid logic program.

For example, if you access computer transfer mode with an HHP that has a logic program for a model 110CPU51201 PLC in its flash, the computer transfer screen looks like this:

```
  COMPUTER   XFER MODE
Computer xfer   <> HHP
a CPU51201*   Program
Continue
```

Selecting a PLC Model Number for the Slave

If you want the master device to download a program to what it thinks is a PLC other than the default, you can select any one of the 14 possible Micro PLC model numbers:

- **CPU31100**
- **CPU31101**
- **CPU31102**
- **CPU31103**
- **CPU41100**
- **CPU41101**
- **CPU41102**
- **CPU41103**
- **CPU51200**
- **CPU51201**
- **CPU51202**
- **CPU51203**
- **CPU61200**
- **CPU61203**

To select any one of these model numbers, scroll through the list by pushing disa ◀ or enab ▶ with the cursor under the displayed model number on the third line.

When you have cursored to the desired model number, push foff ▼ to move the cursor under the **Continue** selection on the bottom line of the screen, then push command enter. A slave status screen such as this one appears:

```
  COMPUTER   XFER MODE
<Enter>   to leave
Panel xfer CPU51201*
ADD=01 9600 RTU 1 EV
```

where the model number that the master perceives the slave to be appears on the third line and the slave's comm port parameters appear on the fourth line. This status screen does not have a cursor, and it cannot be edited—i.e., the model number and comm parameters cannot be changed at this point.

If you want to continue with the upload or download process, issue the appropriate command from the master device. If you want to exit this computer transfer mode screen, push command enter.

Uploading a Program to the Computer

If the HHP is currently holding a logic program in its flash, you can upload that program to the computer (master device). To support this upload at the HHP end, you must select the model number that has an asterisk next to it in the slave status screen.

☞ **Note** An HHP with a valid logic program in its flash always shows that model number followed by an asterisk.

No more than one model in the list of available model numbers can have an asterisk next to it, since the HHP can store no more than one logic program in its flash at any given time.

If none of the model numbers shows an asterisk next to it, then the HHP does not currently store a logic program. In this case, an upload to the master cannot be supported—i.e., panel softwares such as Modsoft Lite do not allow you

to upload unconfigured (null) PLC environments.

The upload operation must be initiated by the master device. For example, if you are using Modsoft Lite on your computer master, and you want to upload a 110CPU51201 program from the HHP to the panel software, follow this procedure:

Step 1. Make sure that the comm port parameters on the serial port of the computer master match those of the HHP slave—i.e., Modbus address 01, 9600 baud, RTU communications, 1 stop bit, and even parity.

Step 2. Select the **`File to PLC`** command from the **`Transfer`** menu on the Modsoft Lite main menu.

Result. The computer master device initiates an upload of the logic program from the HHP to Modsoft Lite. While the upload is running, the red **status** LED on the HHP goes ON to indicate communications in process.

When the upload is complete, the LED goes OFF and a message appears on the master display telling you that a new program has been copied. The slave status screen on the HHP does not change.

Step 3. To exit the slave status screen of the HHP, push command enter on the HHP keypad. The following screen appears on the HHP display:

```
SAVE TO HHP FLASH?
<Yes> = yes
<No> = clear flash
<Exit> = exit
```

Step 4a If you want to get out of computer transfer mode, simply push clr exit .

Result. The HHP returns you to the top-level screen.

Step 4b If you want to clear a stored program out of the HHP's flash, push +net 0 no .

Result. The HHP posts this message on the bottom line of the screen —**`Please wait`**—while the program is being cleared. When the operation finishes, it posts another message—**`Request completed`**.—momentarily before returning you to the top-level screen.

☞ **Note** Initiating a save to HHP flash at the end of an upload is meaningless, since the program is already there. Thus, the IPI 1 yes option on the second line of the screen above does not apply to an upload operation.

Downloading a Program to the HHP Slave

In a master-to-slave download procedure, the computer master device treats the HHP slave as if it was the Micro PLC identified by the model number displayed on the third line of the HHP's slave status screen. The model number you select in the HHP must correspond to the PLC model for which the program in the master has been created.

For example, if you have created a logic program for a 110CPU51200 environment with Modsoft Lite and you want to write that program to HHP flash, then you must select **`CPU51200`** in the HHP before initiating the download.

When you are downloading to the HHP, the HHP's flash may be clear—i.e., there is no asterisk next to any model number that you select—or it may al-

ready have a program in it. If it does have a program in it when a download is initiated, that program will be overwritten by the download when you eventually save the program to HHP flash. You may want to clear the HHP's flash prior to initiating the download to eliminate any possible confusion.

How to Clear a Program from the HHP's Flash

To clear a program from the HHP's flash prior to a download, exit the slave status screen by pushing command enter, then push +net 0 no in the next screen to clear the flash:

```
  COMPUTER   XFER MODE
<Enter>   to leave
Panel xfer CPU51201*
ADD=01 9600 RTU 1 EV
```

command enter

```
SAVE TO HHP FLASH?
<Yes> = yes
<No> = clear flash
<Exit> = exit
```

+net 0 no

```
 HUNTING FOR COMMS
Please  wait . . .
```

After the flash is cleared, the HHP returns to the top-level screen. If you then re-enter computer transfer mode, the default PLC model number, **CPU31100**, appears on the third line of the screen:

```
  COMPUTER   XFER MODE
Computer xfer   <> HHP
a CPU31100     Program
Continue
```

This default model number without a cursor indicates that the HHP now has no program in its flash.

How to Download a Program to the HHP's Flash

The download operation must be initiated by the master device. For example, if you are using Modsoft Lite on your computer master, and you want to download a 110CPU51200 program to the HHP, follow this procedure:

Step 1. Make sure that the PLC model number displayed on the third line of the HHP's slave status screen matches the model number for which the program in Modsoft Lite has been designed.

Step 2. Make sure that the comm port parameters on the serial port of the computer master match those of the HHP slave—i.e., Modbus address 01, 9600 baud, RTU communications, 1 stop bit, and even parity.

Step 3. Select the **PLC to File** command from the **Transfer** menu on the Modsoft Lite main menu.

Result. The computer master device initiates a download of the logic program to the HHP. .

☞ **Note** The download from master to HHP is not direct to the HHP's flash. The master, which is treating the HHP as a programmable controller, sends the program to what it believes is the PLC's user data memory (state RAM).

While the download is running, the red **status** LED on the HHP goes ON to indicate communications in process.

When the download is complete, the LED goes OFF and an asterisk appears beside the PLC model number on the third

line of the HHP's slave status screen. The asterisk indicates that there is now a program in the HHP for the specified PLC model.

Step 4. Exit the slave status screen of the HHP by pushing command enter on the HHP keypad. The following screen appears on the HHP display:

```
SAVE TO HHP FLASH?
<Yes> = yes
<No> = clear flash
<Exit> = exit
```

Step 5. Save the newly transferred program the HHP flash by pushing I/O 1 yes .

Result. The HHP posts this message on the bottom line of the screen —**Please wait**—while the program is saved to flash. When the operation finishes, it posts another message—**Request completed**.—momentarily before returning you to the top-level screen.

Note If you do not save the new program to flash at this point in the process, it will be lost when you exit slave mode.

Changing HHP Executive Firmware

The executive firmware for your Handheld Programmer has been factory-installed in the unit's flash. Updates to the executive may be made to increase functionality and fix bugs.

An executive download procedure is handled in the HHP in much the same way as a logic program download. The key difference between the two procedures is the software being run on the master computer—in this case, a loader utility program rather than a logic programming software.

Updating the Executive with a Loader Utility Program

You can update the executive revision by connecting the HHP in slave mode to a computer that contains a special binary executive software file and a loader utility program. You download the binary executive file from the computer to the HHP utilizing the loader utility.

The loader utility contains five files:

- LOADER.EXE, an executable file that performs the loading function
- LOADER.HLP, a help text file
- LOADER.NDX, an index file for help screens
- MCMIII.MSG, the Modcom III error message file
- README.1ST, a file that explains exactly how to perform the update

The loader utility and the latest executive software can be obtained:

- Via the Customer Service Bulletin Board (24 hours a day, 365 days a year, at no charge)

  From your local Modicon Sales Representative

- As part of a Modicon Micro Flash RAM Subscription Service (Modicon part number SR-FLCH-MCR)

The latest revisions of the HHP and other Modicon executive firmwares are always listed on the Customer Service BBS and on Modfax.

Accessing Modfax

Modfax is an automatic document retrieval system available to Modicon customers located in the U.S. and Canada. The system is self-prompting. To access Modfax, call (800) 468-5342 and select option 3. Have your FAX number available when you call.

For additional hardware or software technical assistance, call the Modicon Field Support Center at (800) 468-5342 or (508) 794-0800 (outside U.S. and Canada) and select option 1.

The relevant Modfax bulletins regarding the HHP include:

- #3726 Executive Updates
- #3727 Available Flash Revisions
- #1113 Bulletin Board Help

Accessing the Customer Service Bulletin Board

The Modicon Customer Service BBS provides several features and benefits (see Modfax Document #1113). BBS members may use the procedure given below or may proceed directly to the `Flash Lib`. Downloading a new HHP executive from the BBS does not cost any credits.

Non-BBS members can use the following procedure to retrieve a binary executive file and the loader utility from the BBS:

Step 1. Using your modem and communication package, dial 508-975-9779. Dial at your modem's maximum baud—we support up to 14,400 baud, no parity, 8 data and 1 stop.

Step 2. If it is your first time calling, you need to create an account—to do this, answer the five questions you will be asked at this time.

Step 3. When you reach the main menu, type

M <enter>

You will be welcomed to the flash RAM download service.

Step 4. Select the number from this menu that corresponds to the HHP. You will get a list of eight files—select the number with the latest HHP rev.

Step 5. Select the download protocol that matches your communication package protocol.

☞ **Note** ZMODEM is the preferred protocol—KERMIT and XMODEM are also good.

Step 6. If your package has ZMODEM, the download commences automatically. With the other protocols, you may need to tell your communications software that you wish to download a file—**<page down>** in PROCOMM, for example—then select the protocol to match the one previously selected on the BBS.

Step 7. You should now have the appropriate file in your download path (determined by your communications package).

Step 8. Let's assume that the first file you take is the binary executive file. To get the loader utility, push **<enter>** once and select **L** from the menu. Then restart this procedure at step 5 above.

Step 9. The downloaded files are compressed and will self-extract when executed.

Step 10 Follow the instructions given in the README.1ST file to update the HHP executive.

HHP Executive Download Procedures

Because the HHP must be connected to a computer to execute the download, it must get its power from an external power supply such as the one provided in the 110VIA19200 Computer Transfer Kit. Connect the power supply to the HHP's power connection on the bottom of the panel.

Connect the RS-232 communications cable first to the HHP and then to the computer, using one of the special 9- or 25-pin adapters available from Modicon.

When the HHP receives power from the external power supply, a **NO COMMS** screen appears when it starts up:

Hhp port	NO COMMS
Slave Hhp	SINGLE
	UNKN PLC

Move the cursor under **Slave hhp** and push command enter. The HHP goes into slave HHP mode, and the following screen appears:

Simpl message	HHP
Computer xfer	SLAVE
Security	Rev
	1.0

Move the cursor under **Computer xfer** and push command enter. The following screen appears:

```
  COMPUTER   XFER MODE
Computer xfer   <> HHP
a CPU31100 *   Program
Continue
```

With the cursor under the PLC model number on the third line of the screen, push disa ◀ to select **FLSHEXEC**. Then push foff ▼ to move the cursor under **Continue** on the bottom line of the screen, and push command enter. The following screen appears:

```
  COMPUTER XFER MODE
<Enter>  to  leave
Panel xfer FLSHEXEC
ADD=01 9600 RTU 1 EV
```

All the information in this screen is status information—the cursor does not appear, and it cannot be edited. From this point on. all the transfer procedures are handled from the computer that has called up the BBS.

Caution When a new executive is loaded to the HHP, all ladder logic, configuration, and I/O map data existing in the HHP will be wiped out. If you want to save this information, you should download it to the PLC before you start the executive download, then upload it from the PLC after the new executive is installed.

At the prompt on your computer terminal, type **LOADER**, then push the **<RETURN>**. Then make sure that your computer's communication parameters correspond with those shown on the bottom line of the HHP screen directly above.

As the HHP executive download proceeds, the following five messages appear on the computer terminal screen sequentially:

- **Waiting for NODE to rejoin link in Kernal mode**
- **Erasing Executive storage area**
- **Programming executive storage area**
- **Waiting for NODE to rejoin link in Normal mode**
- **Download operation successful**

While the computer is downloading the executive to the HHP, the LED on the HHP flashes at three-pulse intervals.

Note If you get an error during the executive download, you will need to restart the download procedure.

Once the download has been successfully completed (after about 15 minutes), the HHP automatically restarts itself and comes up with the **NO COMMs** screen. Remove the AC power from the external power supply adapter, then disconnect the adapter from the HHP. You can then disconnect the RS-232 communication cable from the computer and connect it to a Micro PLC to resume normal usage.

Chapter 8
Using an HHP as a Simple ASCII Messaging Display

- The HHP's Simple Messaging Mode
- ASCII Characters Supported by the HHP
- Using an HHP as a Display Terminal for Simple ASCII Messages
- Using an HHP as an Message Response Tool

The HHP's Simple Messaging Mode

Simple messaging mode is a special case of HHP slave mode where the HHP remains connected to a Micro PLC but becomes a slave to its ladder logic program.

Upon entering simple messaging mode, the HHP adjusts its port parameters to those of the Micro PLC simple ASCII port and de-asserts the DSR line on the port connection.

☞ **Note** See Chapter 2 of ***Modicon Micro Ladder logic Manual*** (GM-MICR-LDR) for more details on the PLC comm port settings.

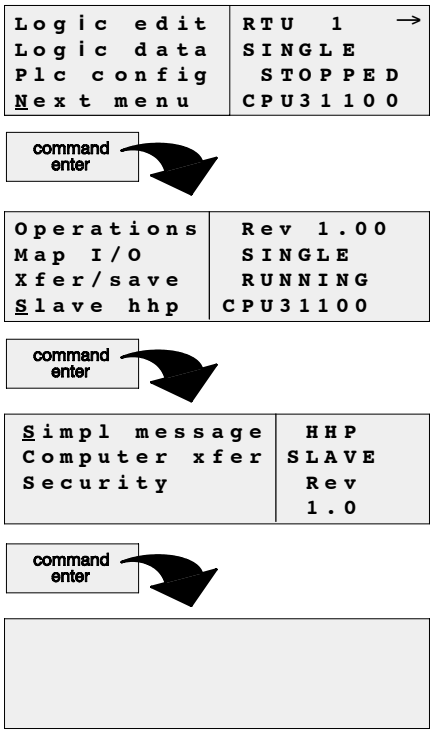
When you put the HHP in simple messaging mode, it displays ASCII characters being sent to it from ladder logic in the PLC on its LCD screen. The ASCII characters are generated by one or more **COMM** instructions executing in ladder logic.

In addition, keys 1 ... 9 and A ... F on the HHP keypad can be pushed to send ASCII characters back to a **COMM** instruction in the PLC logic program—i.e., **COMM** instructions can be set up to receive as well as to send messages.

☞ **Note** For more details concerning the **COMM** instruction, refer to Chapter 8 of the ***Modicon Micro Ladder logic Manual*** (GM-MICR-LDR).

Accessing the Simple Messaging Screen

The slave HHP mode can be accessed in an HHP via the following menu path:



When you select **Simpl message** from the slave HHP menu, the screen goes under the control of the PLC logic program. ASCII character strings can then be sent to the screen from **COMM** instructions executing in ladder logic. The complete list of ASCII characters that can be written to the HHP is shown on pages 149 ... 150.

COMM instructions can also be set up to receive messages from the HHP. The responses from the HHP are restricted to certain keystrokes on the keypad (see page 153).

ASCII Characters Supported by the HHP

The HHP can display any of the following ASCII characters when they are received from the Micro PLC:

ASCII Character	Dec Value	Hex Value
Bell	7	07
Linefeed	10	0A
Formfeed	12	0C
Carriage return	13	0D
→	26	1A
←	27	1B
Space	32	20
!	33	21
"	34	22
#	35	23
$	36	24
%	37	25
&	38	26
'	39	27
(	40	28
)	41	29
*	42	2A
+	43	2B
'	44	2C
-	45	2D
.	46	2E
/	47	2F
0	48	30
1	49	31
2	50	32
3	51	33
4	52	34
5	53	35
6	54	36
7	55	37
8	56	38
9	57	39
:	58	3A
;	59	3B

ASCII Character	Dec Value	Hex Value
<	60	3C
=	61	3D
>	62	3E
?	63	3F
@	64	40
A	65	41
B	66	42
C	67	43
D	68	44
E	69	45
F	70	46
G	71	47
H	72	48
I	73	49
J	74	4A
K	75	4B
L	76	4C
M	77	4D
N	78	4E
O	79	4F
P	80	50
Q	81	51
R	82	52
S	83	53
T	84	54
U	85	55
V	86	56
W	87	57
X	88	58
Y	89	59
Z	90	5A
[	91	5B
]	93	5D
^	94	5E
_	95	5F
a	97	61
b	98	62
c	99	63

ASCII Character	Dec Value	Hex Value
d	100	64
e	101	65
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B
l	108	6C
m	109	6D
n	110	6E
o	111	6F
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A
{	123	7B
|	124	7C
}	125	7D
ü	129	81
ä	132	84
ö	148	94
¢	155	9B
£	156	9C
ñ	164	A4
▮	219	DB
α	224	E0
β	225	E1
Σ	228	E4
σ	229	E5
µ	230	E6
Ω	234	EA
∞	236	EC
ε	238	EE
÷	246	F6

The full set contains 256 ASCII characters. Any of the 256 not shown in this list are not available for display on the HHP.

The Bell Character

Whenever the HHP receives the ASCII bell character (7 decimal, 07 hex) while the unit is in simple messaging mode, the **status** LED is illuminated.

A bell-triggered LED remains ON until you push a key on the HHP keypad. Any key except clr exit can be pushed to turn OFF the LED—pushing clr exit takes you out of the simple message mode, and ‡ clr exit clears the screen of all characters as well as turning OFF the LED.

The Linefeed Character

The linefeed (LF) character (10 decimal, 0A hex) moves the cursor from its current row position to the same position in the row directly below it.

The Formfeed Character

The formfeed character (12 decimal, 0C hex) clears message information from the four lines currently displayed on the HHP screen.

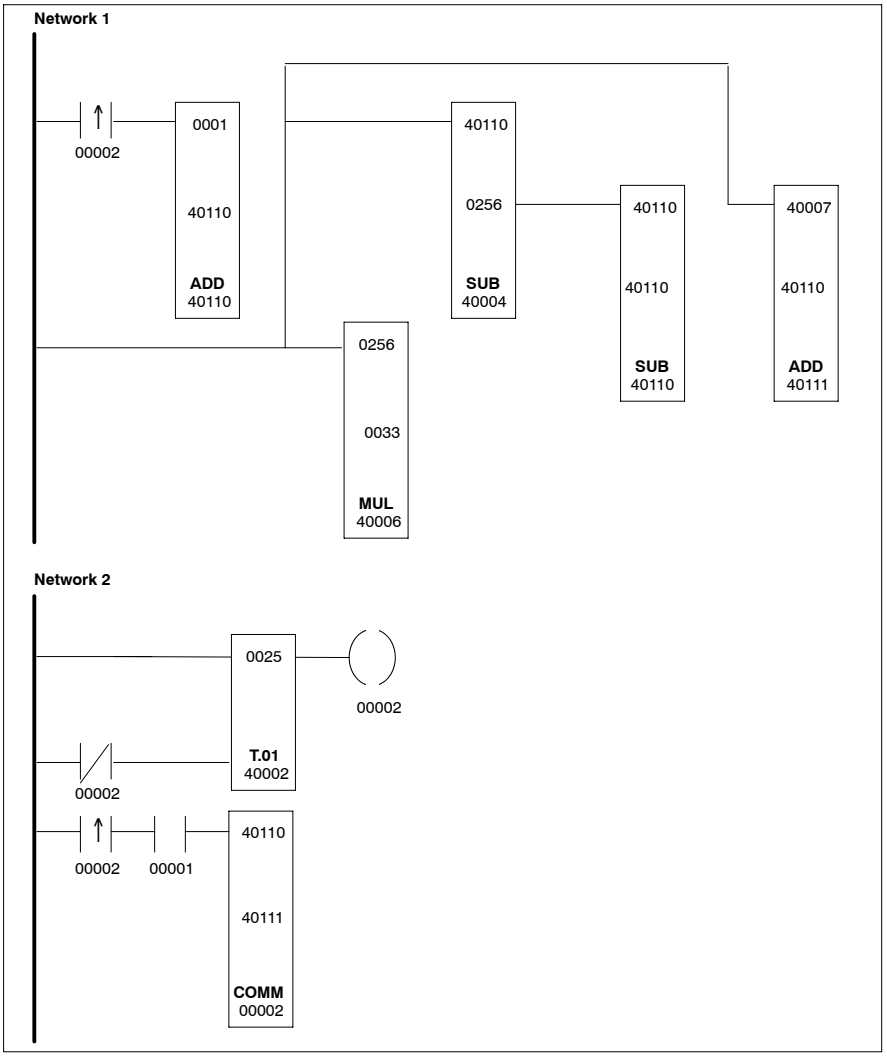
The Carriage Return Character

The carriage return (CR) character (13 decimal, 0D hex) moves the cursor to the leftmost position in the current row.

Using an HHP as a Message Display

A Ladder Logic Example: Displaying the ASCII Character Set on the HHP

Here is a pair of logic networks that will generate a continuous display of all the available ASCII characters. Each character is preceded by an exclamation point (`!n`), and, consequently, each ASCII character that is not available on the HHP is shown as two exclamation points next to one another.



Network 1

Each time that positive transitional contact 00002 in network 1 is fired, it increments the value stored in register 40110 by one. The combination of the two **SUB** instructions in the fourth and fifth rows of network 1 assures that the value in register 40110 clears back to zero when the **ADD**-block increments reach 255. The values 0 ... 255 (00 ... FF hex) in register 40110 correspond to the decimal values of the ASCII characters shown on the previous pages.

Meanwhile, the **MUL** instruction in network 1 performs a straightforward multiplication of 256 x 33. The result, which is stored in the two register 40006 and 40007, is 8448 decimal or 2100 hex. The hex value is the one of interest because the high-byte hex value 21, which will be stored in register 40007, represents the ! ASCII character.

The last **ADD** instruction in network 1 adds the value in register 40007 (the hex representation of !) to the current value in register 40110 and stores the sum in register 40111. As a result, the high byte of register 40111 always contains the hex value 21 and the low byte contains an incrementing value in the range 00 ... FF hex.

Network 2

Network 2 contains a timer that determines the frequency at which contact 00002 (in both networks 1 and 2) will be fired. It is set here to fire the contact every $^1/_4$ second.

Register 40111 is the source register for the **COMM** instruction in network 2, and register 40100 is the first register in the **COMM** control block. The control block is set up in this way:

Register	Value	Meaning
40100	1110	Write ASCII character, no CR/LF
40101	0000	N/A
40102	0002	Transmit two ASCII characters
40103	0000	N/A
40104	0000	N/A
40105	0001	Local port used for messaging
40106	0000	N/A
40107	0000	N/A
40108	0000	N/A
40109	0000	N/A

You can enter these networks into ladder logic with your HHP to demonstrate the message display capabilities of the device. As these two networks execute in the logic scan, each available ASCII character will be displayed sequentially on the HHP screen. Each supported character will be preceded by an exclamation point.

When disabled coil 00001 is ON, it enables the transmission of ASCII characters. To stop the flow of characters, turn coil 00001 OFF.

Using an HHP as an Message Response Tool

The following 15 keys (**A** ... **F**, **1** ... **9**) on the HHP keypad can be used to create message responses back to a **COMM** instruction in the PLC logic:

Data Entry Keys

vopn E	I F	move 7	mtrx 8	othr 9
( ) C	− D	cntr 4	timr 5	math 6
| | A	\/| B	|P| 1 yes	|N| 2	(M) 3

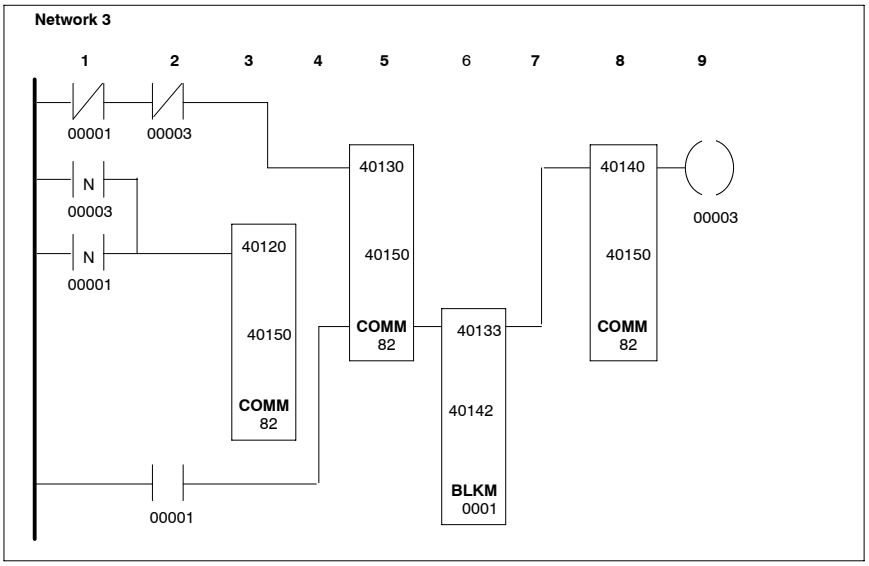
Five other keystroke sequences can be used as control keys in an ASCII message response:

Message Control Keys

command enter	Sends a CR on the wire and places the cursor at the beginning of the next line
‡ command enter	Places the cursor at the beginning of the next line; nothing goes out on the wire
‡ cir exit	Clears the display; nothing goes out on the wire
disa ◄	Returns the cursor to the beginning of the current display line; nothing goes out on the wire
foff ▼	Places the cursor on the next line; nothing goes out on the wire

A Ladder Logic Example: Responding to the PLC

On the following page is a ladder logic network that allows you to send a message to the PLC from the HHP keypad. The message is then echoed back to you on the screen to indicate that it has been received.



Network 3

Normally closed contact 00001 stops the activity of networks 1 and 2 (page 151) when it is OFF.

Normally closed contact 00003 starts the second **COMM** instruction (in column 5) in network 3. As can be seen by the values set up in the control block below, this **COMM** instruction is designed to receive messages entered onto the HHP keypad:

Register	Value	Meaning
40130	1020	Read ASCII character, CR or LF
40131	0000	N/A
40132	0082	A message can contain up to 82 ASCII characters
40133	0000	Stores the actual number of characters received
40134	0000	N/A
40135	0001	Local port used for messaging
40136	0000	N/A
40137	0000	N/A
40138	0000	N/A
40139	0000	N/A

The received string of ASCII characters is completed when 82 characters have been sent or when the operator pushes command enter .

The received message characters are stored in register 40150, in the middle-node *destination* buffer in the second **COMM** instruction.

The third **COMM** instruction (in column 8) in network 3 then echoes the received characters back to the HHP screen. The number of characters received by the second **COMM** instruction—i.e., the value stored in the fourth register of its control block, 40133—is block moved into the third register of the third **COMM** instruction's control block to define the send-message length. Register 40150, the destination buffer of the second **COMM** instruction, becomes the *source* buffer for the third **COMM** instruction.

The control block for the third **COMM** instruction is set up like this:

Register	Value	Meaning
40140	1120	Write ASCII character
40141	0000	N/A
40142	00*nn*	The number of characters received from the HHP, from register 40133
40143	0000	N/A
40144	0000	N/A
40145	0001	Local port used for messaging
40146	0000	N/A
40147	0000	N/A
40148	0000	N/A
40149	0000	N/A

The other logic circuit, implemented with the two contacts and the first **COMM** instruction (in column 3) in network 3, is used for housekeeping. When coil 00003 transitions from ON to OFF, this logic clears out the message receive buffer in the Micro PLC, just in case there has been a comm error logged there.

If, for example, you have a logic network such as network 3 in your program, you can enter **984** followed by command enter from your HHP keypad while you are in the simple messaging editor:

```
9 8 4
```

The three-character message (the carriage return is transparent) is sent to the second **COMM** instruction and the values are stored in destination register 40150. The received message is then moved to the third **COMM** instruction, which sends the message back to the HHP display; register 40150 now becomes the source register:

```
9 8 4
9 8 4
```

Here are the values for the control block of this **COMM** instruction:

Register	Value	Meaning
40120	1000	Flush input buffer
40121	0000	N/A
40122	0000	N/A
40123	0000	N/A
40124	0000	N/A
40125	0000	N/A
40126	0000	N/A
40127	0000	N/A
40128	0000	N/A
40129	0000	N/A

Chapter 9
The HHP's Optional Password Security System

- The Four Levels of Password Security
- Accessing the Security Scheme
- Enabling Password Security
- Disabling Keyswitch Security
- Assigning Passwords

The Four Levels of Password Security

If your application requires that you limit access to certain editing capabilities of the HHP, an optional password security scheme is provided. The password scheme needs to be set up by an administrator who can limit HHP/PLC access to four levels of user.

Accessing a Security Level without a Password

When password security is enabled, the password fields are tested whenever the PLC is powering up. The PLC comes up accessible to users at the highest level that has a null password. For example, if the administrator has set a password for level 3 security but has not set a password for level 2 security, a user will be able to access level 2 privileges without knowing a password. If both level 2 and level 3 are password-protected, then the user can access only level 1 privileges without using a password—level 1 security is always unprotected.

Level 1 Security Privileges

When password security is enabled, all users have level 1 security privileges. Level 1 gives a user access to a limited set of capabilities in the HHP, and these capabilities cannot be password-protected.

Level 1 Privileges

- Can view the current configuration parameters
- Can view the current I/O map parameters, including comm health and comm error statistics
- Can view the ladder logic program and monitor power flow in logic edit mode
- Can view data displayed in the reference data screen
- Can transfer user logic from the PLC or from another panel—e.g., Modsoft Lite—to the HHP and verify the transfer
- Can transfer user logic to another panel—e.g., Modsoft Lite—from the HHP
- Can transfer a new HHP executive to the Hand-held Programmer
- Can operate in messaging and computer slave modes

Level 1 Restrictions

- Cannot change the PLC's configuration (including the inability to execute a new autoconfig)
- Cannot write any new parameters to the PLC's I/O map
- Cannot start or stop the PLC in operations mode
- Cannot clear user logic in operations mode
- Cannot transfer user logic from the HHP to a PLC
- Cannot save user logic to the PLC's flash memory
- Cannot edit any discrete or register reference values
- Cannot edit the ladder logic program

Level 2 Security Privileges

Additional HHP capabilities can be accessed by a user with level 2 security privileges. Level 2 privileges can be password-protected by the admin so that users without knowledge of the password are locked out. If a level-2 password is not assigned, then all users can access these capabilities.

Level 2 Privileges

- Can access all level 1 privileges
- Can start and stop the PLC in operations mode
- Can edit discrete and register reference values

Level 2 Restrictions

- Cannot change the PLC's configuration (including the inability to execute a new autoconfig)
- Cannot write any new parameters to the PLC's I/O map
- Cannot clear user logic in operations mode
- Cannot transfer user logic from the HHP to a PLC
- Cannot save user logic to the PLC's flash memory
- Cannot edit the ladder logic program

Level 3 Security Privileges

Level 3 security access is to all HHP and PLC functions except the ability to change the security system—i.e., to all HHP capabilities except admin.

Level 3 privileges can be password-protected by the admin so that users without knowledge of the password are locked out. If a level-3 password is not assigned, then all users can access these capabilities.

Admin Security Privileges

Access to the admin security level requires knowledge of the admin password. The HHP is shipped with a default admin password—**12345**—which may be customized by the admin user.

Level Admin security access allows a user to:

- Install and edit security passwords
- Enable/disable password security
- Enable/disable keyswitch security

To use other HHP and PLC functions, you need to exit the admin level and access level 3 security by entering the level 3 password.

Password Construction

A password comprises any combination of one ... eight alphanumeric characters available on the HHP keyboard—i.e., the letters **A**, **B**, **C**, **D**, **E**, and **F** and the numbers **0** ... **9**.

If your password is less than eight characters long, the HHP automatically fills in the leading characters with zeros. You do not need to enter leading zeros when you type your password.

Only a user with level 4 admin access can create a password.

Accessing the Security Scheme

From the top-level screens in the HHP, the access path to the security screens is as follows:

```
Logic edit   | RTU   1   →
Logic data   | SINGLE
Plc config   |  STOPPED
Next menu    | CPU31100
```

command enter

```
Operations   |
Map I/O      |  SINGLE
Xfer/save    |  RUNNING
Slave hhp    | CPU31100
```

command enter

```
Simple msging |  HHP
Computer xfer | SLAVE
Security      |  Rev
              |  1.0
```

command enter

```
!!!!! SECURITY !!!!!
Enter Password
Install Passwords
Status (pw DISABLED)
```

Monitoring Security Status without Admin Privileges

The HHP gives a user limited access to status inside the security screens if that user does not enter the admin password. If you move the cursor down to **Stat (pw DISABLED)** and push command enter, the following screen appears:

```
-
 KEYSWITCH ENABLED
 PASSWORDS DISABLED
```

The cursor appears in the upper left corner of this screen and it cannot be moved onto either of the two line entries—i.e., the current state of the keyswitch and the password security schemes can be viewed but not changed. To return to the first security screen, push clr exit.

If you move the cursor under **Install Passwords** and push command enter while password security is disabled, the HHP returns an error message on the bottom line of the screen:

```
!!!!! SECURITY !!!!!
Enter Password
Install Passwords
E-PASSWORDS DISABLED
```

Push the fon ▲ or foff ▼ key to clear the error message from the screen.

Becoming Admin

You need admin privileges before you can edit any parameters inside the password security scheme. To become admin, place the cursor under **Enter Password** and push command enter. The following screen appears:

```
!!!!! SECURITY !!!!!
====================
   Enter Password
      >       _<
```

If this is the first time the security scheme is being accessed, a default admin password—**12345**—is in effect. Enter the following sequence:

```
!!!!! SECURITY !!!!!
====================
   Enter Password
     >00012345<
```

```
!!!!! SECURITY !!!!!
====================
You now have ADMIN
  Security Access
```

```
!!!!! SECURITY !!!!!
Enter Password
Install Passwords
Status (pw DISABLED)
```

You are now back at the first security screen, but this time you have admin privileges. These privileges allow you to:

- Enable the password security scheme
- Disable the security keyswitch when it is set in monitor mode, thereby enabling program mode on the HHP regardless of the keyswitch position
- View and edit the current password scheme

Enabling Password Security

A use with admin privileges can enable the security password scheme via this three-step procedure:

Step 1. Move the cursor under **Status (pw DISABLED)** on the first security screen and push command enter .

Result The following screen appears:

```
KEYSWITCH  ENABLED
PASSWORDS  DISABLED
```

Step 2. Move the cursor under **PASSWORDS DISABLED**, push disa ◄ or enab ► to toggle to **PASSWORDS ENABLED**:

```
KEYSWITCH  ENABLED
PASSWORDS  ENABLED
```

Step 3. Push cir exit .

Result You have enabled the password security scheme, and your level of security access will be noted in various screens throughout the HHP system. The first security screen shows the enabled status on the bottom line:

```
!!!!! SECURITY !!!!!
Enter Password
Install Passwords
Status (pw ENABLED)
```

When you exit the security screens, the next screen that appears tells you your current security access level. It is noted in parentheses beside the **Security** entry on the third line:

```
Simple msging  | HHP
Computer xfer  | SLAVE
Security (1)   | Rev
               | 1.0
```

When you exit the security screens as admin, you are always assigned the highest unprotected security access.

In the sample screen above, the user has level 1 security privileges upon exiting the security screens. This indicates that passwords have been assigned to level 2 and level 3 security; the user needs to know those passwords in order to access level 2 and level 3 security privileges.

If you exit the security screens as admin and level 2 and level 3 passwords have not been assigned, then you will exit with level 3 security privileges.

If you exit the security screens as a level 2 user, you will have level 2 privileges throughout the HHP system. If you exit the security screens as a level 3 user, you have full access to all non-admin capabilities of the HHP.

Disabling Keyswitch Security

A user with admin privileges can override the monitor-mode setting of the HHP's security keyswitch—i.e., enable program mode capability while the key is in monitor-mode position. This function can be accomplished while the password security scheme is either enabled or disabled.

To override the keyswitch, follow this three-step procedure:

Step 1. Move the cursor under **Status (pw DISABLED/ENABLED)** on the first security screen and push command enter .

Result The following screen appears:

```
KEYSWITCH  ENABLED
PASSWORDS  DISABLED
```

Step 2. Move the cursor under **KEYSWITCH ENABLED**, and push disa ◄ or enab ► to toggle to **KEYSWITCH DISABLED**:

```
KEYSWITCH  DISABLED
PASSWORDS  DISABLED
```

Step 3. Push clr exit .

Result The HHP will operate in program mode now, regardless of the position of the security keyswitch.

Assigning Passwords

There are two prerequisites for assigning security level passwords in the HHP:

- The user must have admin privileges
- The password security scheme must be enabled

To assign passwords, or simply to enter a display to see the current password assignments, follow this sequence:

```
!!!!! SECURITY !!!!!
Enter Password
Install Passwords
Pass stat (ENABLED)
```

command enter

```
LEVEL2 PSWD
LEVEL3 PSWD
ADMIN  PSWD 00012345
```

The screen directly above shows the initial default password assignments—the admin password is **12345**, and the level-2 and level-3 password are null. At this point, all users have access to all HHP functionality except admin privileges.

As admin, you have the ability to create passwords for level 2 and level 3, and you may also change or delete the admin password.

Password Hierarchy

The password security scheme imposes a set of priorities on the creation of level 2 and level 3 passwords:

- A level 2 password cannot be created unless a level 3 password already exists
- A level 3 password cannot be created unless an admin password already exists

For example, if you start out with the default admin password and no level 2 or level 3 passwords installed and immediately move the cursor up to the top line to enter a level 2 password, the HHP will not accept that password when you try to enter it. An error message appears on the bottom line of the screen:

```
LEVEL2 PSWD 00DEF002
LEVEL3 PSWD
ADMIN  PSWD 00012345
```

command enter

```
LEVEL2 PSWD
LEVEL3 PSWD
ADMIN  PSWD 00012345
E-Lowr lvl paswd set
```

You must move the cursor down to the second line and enter a level 3 password before the level 2 password can be accepted.

Creating a Level 3 Password

Before you begin to create a level 3 password, make sure that an admin password has already been created. Then move the cursor to the second line of the screen—the field on the right side of the screen may be empty (if no level 3 password currently exists) or it may contain a character string (if there is already an existing level 3 password).

To create a new password, or overwrite the old one, type in a character string that consists of 1 ... 8 alphanumeric characters from the HHP keypad. If you type less than eight characters, the HHP will fill all leading characters in the field with zeros. If you are overwriting an old password, the old character

string will disappear as soon as you start typing the new one.

For example, if you want level 3 password to read **DEF003**, type in the string:

```
LEVEL2  PSWD
LEVEL3  PSWD  00DEF003
ADMIN   PSWD  00012345
```

If you have just overwritten an old password (before pushing command enter) and decide you want to restore it, push ‡ cir exit . The previous password will be restored.

If you want to exit the password screen at this point (and save the new password as you exit) push cir exit . If you want to save the password and edit other passwords in the screen, push command enter , then move the cursor into the field of another password you want to edit. Once cir exit or command enter has been pushed, the previous password cannot be restored.

Creating a Level 2 Password

Before you begin to create a level 2 password, make sure that a level 3 password has already been created. Then move the cursor to the top line of the screen—the field on the right side of the screen may be empty (if no level 2 password currently exists) or it may contain a character string (if there is already an existing level 2 password).

To create a new password, or overwrite the old one, type in a character string that consists of 1 ... 8 alphanumeric characters from the HHP keypad. If you type less than eight characters, the HHP will fill all leading characters in the field with zeros. If you are overwriting an old password, the old character string will disappear as soon as you start typing the new one.

For example, if you want the level 2 password to be **DEF002**, type in the string:

```
LEVEL2  PSWD  00DEF002
LEVEL3  PSWD  00DEF003
ADMIN   PSWD  00012345
```

If you have just overwritten an old password and decide you want to restore it, push ‡ cir exit . The previous password will be restored.

If you want to exit the password screen at this point (and save the new password as you exit), push cir exit . If you want to save this password and edit other passwords in the screen, push command enter , then move into the field of another password you want to edit. Once cir exit or command enter has been pushed, the previous password cannot be restored.

Customizing the Admin Password

As the system administrator, you may want to change the admin password to a string that only you know. To overwrite the default admin password, simply push foff ▼ to move the cursor to the third line, then type in the desired character string, again an alphanumeric string of 1 ... 8 characters available on the HHP keypad. For example, if you

want the admin password to be **DEF0AD**, type in the string:

```
LEVEL2  PSWD  00DEF002
LEVEL3  PSWD  00DEF003
ADMIN   PSWD  00DEF0AD
```

If you have just overwritten an old password and decide you want to restore it, push ‡ cir exit . The previous password will be restored.

If you want to exit the password screen at this point (and save the new password as you exit), push cir exit . If you want to save this password and edit other passwords in the screen, push command enter , then move into the field of another password you want to edit. Once cir exit or command enter has been pushed, the previous password cannot be restored.

☞ **Note** As the system administrator, you are responsible for knowing the current admin password at all times. You can reference the above screen to see the other current password listings, but you must know the admin password in order to access this screen.

Deleting a Password

The HHP also imposes a set of priorities on the deletion of admin and level 3 passwords:

- A level 3 password cannot be deleted while a level 2 password exists
- An admin password cannot be deleted while a level 3 password exists

For example, if you have installed all three passwords and you attempt to delete the level 3 or admin password, the HHP will not accept that deletion when you try to execute it. The same error message appears on the bottom line of the screen:

```
LEVEL2  PSWD  00DEF002
LEVEL3  PSWD  00DEF003
ADMIN   PSWD  00012345
```

‡ del srch

```
LEVEL2  PSWD  00DEF002
LEVEL3  PSWD  00DEF003
ADMIN   PSWD  00012345
E-Lowr lvl paswd set
```

To delete a password, move the cursor under the rightmost character of the password you want to delete and push ‡ del srch . Once a password has been deleted, it cannot be restored unless you rekey it.

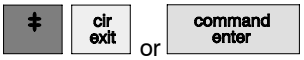
Appendix
HHP System Messages

This appendix provides an alphabetical listing of the system messages generated by the HHP along with an explanation of the messages and, where appropriate, the user response to the condition indicated by the message.

Message	Meaning	User Action
Auto config first	User trying to customize the configuration of an unconfigured PLC	Autoconfig the PLC before customizing its configuration.
Bad ref nbr	Reference number requested in logic data screen is out of range or not valid—e.g., **09999**, **49999**, **19999**, **39999**, **20001**, etc; results from using botm next or direct number entry to exceed the configured number of a given reference type	
Close col disallowed	Cursor is in a column that is not empty	Move cursor to an empty row and re-issue **Close col** via the ‡ command enter menu.
Close row disallowed	Cursor is in a row that is not empty	Move cursor to a column that has no logic elements in it and re-issue **Close row** via the ‡ command enter menu.
Disabled, not used	Search for disabled discrete finds one that is disabled but not used directly in the logic; it may have been disabled via logic data screen operations	
E-00017, coils used	Example of a message that appears when a user tries to imply coils in a node when those coils are used directly or implied in another node	
E-coil 00555 used	Example of message that appears when a user tries to use a coil that is already used	Look for implied coils in the instruction nodes

Message	Meaning	User Action
E-coil not disab	Modbus report—trying to force ON or force OFF a coil, (or input discrete) while it is enabled	
E-coil(s) used	Modbus error report—one or more requested coils already programmed in another place	
E-CRC error	Noisy comms causing loss of one or more bits of data in the packet	May require retrying the operation or program step, depending on nature of comms.
E-disab not allwd	User attempting to disable a coil when not logged in	
E-illegal config	Modbus report—PLC has detected an illegal configuration value	Reconfigure and repeat. Check the configuration with the views—may require software like Modsoft Lite to reconfigure.
E-Illegal operation	Message occurs when user tries to program a node or instruction when that operation cannot be allowed. Examples: • Putting an element other than a coil in column 11 • Putting any node to right of a coil • Replacing a contact with a coil • Replacing a contact or coil with an instruction • Replacing dissimilar sized instructions • Placing a vertical short in column 11 • Placing a vertical short in row 7	

Message	Meaning	User Action
E-illeagl value	System communication error	Depending on where and when it happens, may require fix-up to a logic network or to the I/O map. Check the current network or I/O map after such an error.
E-Incompat programs	User trying to transfer a program for a PLC that is incompatible with the target PLC—e.g., 311/411 to 512/612, 512/612 to 311/411	
E-Inp & Out Incompat	Trying to configure a bidirectional I/O location with combined register and discrete references	Configure the location with register in/out references (3*x* and 4*x*) or discrete in/out references (1*x* and 0*x*).
E-invalid addr	System communication error	Depending on where and when it happens, may require fix-up to a logic network or to the I/O map. Check the current network or I/O map after such an error.
E-invalid cmd	System communication error	Depending on where and when it happens, may require fix-up to a logic network or to the I/O map. Check the current network or I/O map after such an error.
E-invalid insert	System communication error	Depending on where and when it happens, may require fix-up to a logic network. Check the current network after such an error.

Message	Meaning	User Action
E-invalid node type	An attempt to insert a node type not supported by the Micro	Change or delete the node. Look for inconsistent use of references in certain types of instructions—e.g., a 3*x* in the top node of an **EMTH** instruction with the bottom node set to a subfunction that does not support integer input (see the ***Modicon Micro Ladder Logic Manual***, GM-MICR-LDR for details).
E-invalid param	System communication error	Depending on where and when it happens, may require fix-up to a logic network or to the I/O map. Check the current network or I/O map after such an error.
E-key not active	The key pressed is not active at this screen	May be caused if you try to use a non-numeric key when data entry is started—push  or to clear the message
E-Lowr lvl paswd set	Admin's attempt to change a password creates a hole in the levels scheme—e.g., trying to create a level 2 password without a pre-existing level 3 password	

Message	Meaning
E-Modbus error=*xxxx*	Modbus report—nonverbose messages for unexpected Modbus errors, where ***xxxx*** has the following meanings:

xxxx	**Meaning**
0001	A parameter to modcom had illegal value (HHP firmware bug)
0003	Modcom does not support function (HHP firmware bug)
0004	An expected response was not obtained
0005	User-initiated communications abort
0010	An invalid communications mode
0011	An invalid communications mode
0012	Bad comm port—the requested number is out of range
0030	Invalid Modbus address—must be in the range 1 ... 247
0031	Invalid channel—i.e., not a valid port address
0032	Invalid PLC type specified
0033	Illegal broadcast
0034	Attempt to read packed coils across a 257 boundary
0040	Illegal reference start or size specification
0041	Illegal subfunction code (an HHP firmware bug)
0101	Function code not supported by PLC (an HHP firmware bug)
0102	PLC has rejected start address as illegal
0103	PLC has rejected transferred data as illegal
0104	PLC has failed to respond, or abortive error
0105	PLC has returned an ACK exception—request to POLL
0106	PLC busy—unable to service request
0107	PLC has returned an NAK—cannot understand request
0108	PLC has returned a memory parity error
0109	PLC returning an undefined exception error (May be hardware problem in PLC)
0110	Host has received an invalid response buffer
0111	No start of response
0112	Invalid ASCII character in response
0114	Bad address or function code used (an HHP firmware bug)
0116	Unexpected response length received (an HHP firmware bug)
0129	No start of text on program response
0144	Invalid NAK—response
0145	Bad program command checksum
0146	Length has been truncated

Message	Meaning

xxxx	**Meaning**
0160	Invalid command
0161	Invalid address or address range
0162	Memory protect ON
0163	Memory is full
0164	Invalid node type
0165	Coil is used
0166	Invalid reference number range
0167	Invalid parameter
0168	PLC running
0169	Not logged in—a user logged in on another port
0170	Coil not disabled
0171	Search has failed
0172	Illegal configuration
0173	Illegal message
0174	Illegal sequence of insertion
0175	Time base for constant sweep missing
0176	Coil disable not allowed
0177	Cannot modify flash memory
0178	Coprocessor failure
0190	No ASCII message with this number
0191	ASCII message number previously used
0195	Unable to allocate enough space
0196	Executive failure
0199	Slave did not respond before time-out
0499	Invalid parameters to command
0501	Link status: illegal function or orphan poll
0504	Link status: PLC has failed to respond or an abortive error received
0506	Link status: PLC busy—function cannot be performed at this time
0511	Link status: Message size too big for the PLC
0512	Link status: new message started before previous message has completed
0513	Link status: Packet out of sequence—command aborted
0514	Link status: Major sequencing error detected—link reset required
0515	Link status: response is too large for PLC—command aborted
0516	Link status: PLC has returned a reserved response—unrecognized response from PLC

Message	Meaning
E-Must be ADMIN	Unauthorized user trying to perform a password security function that requires admin access
E-No hhp program	HHP has no valid program in flash when the user tries to initiate a transfer from HHP to PLC
E-not disabled	User trying to force a discrete ON or OFF while it is enabled
E-net not found	Network requested by get is not in the PLC
E-PAB inputs > 256	User attempting to address more than 256 input points for A120 I/O expansion in the I/O map editor
E-PAB outputs > 256	User attempting to address more than 256 output points for A120 I/O expansion in the I/O map editor

☞ **Note** The five fixed locations are immune from over programming the I/O points.

E-PLC memory full	No more networks or nodes may be added to the PLC

Message	Meaning	User Action
E-PLC running	Modbus report—PLC running and cannot accomplish the requested task	If necessary, stop the PLC and repeat the function.
E-ref 10017, type	Example of a bad reference number by virtue of type—a 1*x* is not allowed in an I/O map output field	
E-ref 49999, range	Example of a bad reference number by virtue of range—not enough 4*x* registers configured to cover the reference number requested	Use an in-bounds reference
E-ref out of range	Modbus report—user trying to read or write a reference number that is beyond configured range	
E-Simpl mesage param	PLC's simple message parameters outside of the range supported by the HHP: • 1200/2400/4800/9600 baud • 7/8 bits • 1/2 stop bits • even/odd/no parity This error prevents the HHP from entering simple messaging mode	Change the simple message parameters in the **set config** screens to a value that the HHP supports
E-Timeout, comm down	PLC comms have timed out	Check for an open cable, a very weak or intermittent connection, a very noisy environment, or other hardware problems.
E-Trans failed	PLC program is too large to fit in the HHP or HHP program too large to put in the PLC	

Message	Meaning
Empty network	User entered an empty network via botm next , top prev , or back get , or after inserting or deleting a network via ⇞ command enter
End of user logic	User pushed botm next in logic edit screen when the cursor is in the last logic network
Enter segment *xx*	User pushed botm next or top prev and crossed a segment boundary; *xx* is a number from 1 ... 32 (A different message is seen when the user crosses into the last segment, which is reserved for subroutine logic.)
Enter subrtine area	Equivalent to **Enter segment *xx***, where ***xx*** is the last segment in the logic program
Field not empty	User has pushed back get on a logic data field that already has a reference
Hhp in monitor mode	Keyswitch in the monitor mode position with its functionality enabled
Hunting for comms Please wait...	HHP currently attempting to communicate with the PLC by using preassigned addresses and baud rates and modes
Hunting for comms Please wait... 123	HHP is currently attempting to communicate with the PLC by using all legal addresses 1 ... 247; this could take on the order of 11 minutes (worst case)—the reported number after the **Please wait** message indicates the address being tried.
Illegal child nbr	User trying to select a child in a parent I/O map when the child number given is unconfigured or zero
Locations 1-5 only	User trying to change to an I/O location outside the range 1 ... 20 in a PLC that supports A120 I/O expansion, or outside the range 1 ... 5 in a PLC that does not support A120 I/O
Need Security lvl *x*	When password security is enabled, your current security level is too low for access or for unlimited access to a function; ***x*** is the required level for unlimited access
Network not found	Network being searched for is not in user logic program

Message	Meaning	User Action
Net Unsolvd	Network on screen is unsolved by virtue of being skipped or, in the case of a subroutine, not being executed at this time	Clear the message by pushing a key—it does not get cleared by a subsequent solving of the network.
No more	Search for subsequent disabled discretes using botm next finds no more disabled 0*x* or 1*x* references	
No ref to get	User has pressed back get in logic data when there is no node at the cursor position in the current network; also results from pushing top prev on a reference at its lowest range	Push subr data to toggle back and forth between logic edit and logic data. Move cursor in the logic network to a valid reference.
No trace	Fails to trace to a coil in user logic—no coil corresponding to the contact is explicitly used in the logic program	
None	Search for disabled discretes using back get finds no disabled 0*x* or 1*x* references	
Not found	Object of a search is not in the user logic	Object of the search should still be visible on the screen
Open col disallowed	• Column 11 has one or more coils • Column 10 has one or more nodes other than coils	

Message	Meaning	User Action
Open row disallowed	• Row seven not empty • Row six has one or more vertical shorts • Cursor is on a row that intersects an instruction's middle or bottom node	
Other comm attached	Login to the PLC fails because another panel is logged into the other comm port	
PASSWORDS DISABLED	Admin must enable password security before trying to install or modify password characters; also results from entry of non-admin password when password protection is disabled	
PLC in kernel mode	PLC has no valid executive program	Download a new configuration and ladder logic program; if you still cannot start the Micro, load a new PLC executive to the PLC flash; if the problem still persists, you have a hardware problem in the PLC
PLC in optimize mode	Another programming panel—e.g., Modsoft Lite—has left the PLC in optimized mode; logic editing is not allowed and power flow cannot be viewed	
PLC NOT CONFIGURED	Attempting to transfer an unconfigured PLC environment to an HHP	
PLC NOT KNOWN	User trying to transfer a PLC program to an HHP that is not on the list of supported models	

Message	Meaning	User Action
PLC running	**Start** is selected from the **Operations** menu when the PLC is already running; this message may also appear at other places as a reminder that full programming capability is not available because the PLC is running	
PLC stopped	**Stop** is selected from the **Operations** menu when the PLC is already stopped	
Ref 10001 used S/01	Sample warning message for duplicate use of reference (**10001**) in the self group, LOC01	
Ref 30001 used C1/1	Sample warning message for duplicate use of reference (**30001**) in the child 1 group, LOC01	
SAVING FAILED	Save within PLC operation not successful	Check PLC hardware and cable system
SAVING SUCCESSFUL	Save within PLC operation was successful	
Strt of user logic	User pushed top prev in logic edit screen when the cursor is in logic network 1	
Subroutine not found	Subroutine search initiated with the cursor on a **JSR** finds no associated **LAB** in the logic	
User logic cleared	Logic successfully cleared via the **Operations** menu	

Message	Meaning
`Xfer to hhp complete`	PLC's user logic has been successfully transferred and stored in HHP flash
`Xfer to PLC complete`	The logic program stored in the HHP flash has been successfully transferred to the PLC